(2)

COPY

AD-A217 080

# A Cognitive Model of Human-Computer Interaction in Naval Air ASW Mission Management


Wayne Zachary, Ph.D.
Joan M. Ryder, Ph.D.
Monica C. Zubritzky

CHI SYSTEMS
Gwynedd Plaza III
Bethlehem Pike and Sheble Lane
Spring House, PA 19477



CHI SYSTEMS TECHNICAL REPORT 891215.8704
15 December, 1989

CONTRACT N00014-87-C-0814

R&T Project Code: 4429007

# A Cognitive Model of Human-Computer Interaction in Naval Air ASW Mission Management

Wayne Zachary, Ph.D.
Joan M. Ryder, Ph.D.
Monica C. Zubritzky

CHI SYSTEMS
Gwynedd Plaza III
Bethlehem Pike and Sheble Lane
Spring House, PA 19477

CHI SYSTEMS TECHNICAL REPORT 891215.8704
15 December, 1989

CONTRACT N00014-87-C-0814

R&T Project Code: 4429007

## Table of Contents

## List of Figures & Tables

## ACKNOWLEDGEMENTS

We wish to acknowledge the important contributions to this research by the TACCOs stationed at the Naval Air Development Center and NAS Willow Grove who participated as experimental participants in this research. Without their enthusiasm and generous contribution of their own time, this research would not have been possible. They are not acknowledged by name only because their anonymity as experimental participants was promised. The authors also wish to thank Mr. Stanley Winsko (Naval Air Development Center) and LT Michael Holmes (USN Aerospace Medical Service Corps) in reviewing and commenting on our progress, and in identifying and gaining access to experimental participants, and of Mr. Eric Haas of Analytics in providing ASW expertise and advise. In addition, valuable comments and suggestions have been provided by Dr. John O'Hare of ONR, Mr. William Weiland of the University of Maryland, Dr. Lee Goodman of Magnavox Electronics (formerly of NADC), and Dr. Stanley Schwartz here at CHI Systems.

# 1. INTRODUCTION AND BACKGROUND

This paper describes on-going research to investigate the cognitive basis for human-computer interaction and decision-making in complex, real-world environments, particularly those which unfold in real-time and make multiple demands on the attention of the human decision-maker. The main emphasis in the project is to explore the extent to which a model of the person's problem-solving strategies in these real-time multi-tasking (RTMT) environments can lead to the design of effective human-computer interfaces for them. RTMT environments include many of the most challenging problem domains humans face, such as aircraft (and other vehicle) cockpits, nuclear power control rooms, automated manufacturing environments, air traffic control, hospital operating rooms, satellite and telecommunication network control, and weapons systems operation, to name but a few. These problem environments are undergoing rapid computerization, and are all critical to economic, personal, and national well-being, and are thus inherently worthy of close study.

There are several goals in this project. The first is to develop new cognitive and human-computer interaction modeling methodologies and tools. Most of the existing tools and theories available for analyzing and understanding cognitive processes in human-computer interaction are not directly applicable to real-time or multi-tasking domains, because most previous research has focused on synthetic or highly constrained non-RTMT tasks (such as text editing). The second goal is to model both canonical aspects and primary individual variations of the human-computer interaction in a specific, realistic RTMT domain. This is done to provide a basis for testing and refining the modeling languages and tools developed here. The third goal, which is not discussed in this paper, is to use the model of human-computer interaction in the example domain to design and implement improved human-computer interfaces in the domain.

The RTMT cognitive modeling tools described here were developed from an integration of two existing frameworks, the GOMS notation of Card, Moran and Newell (1983) and the Blackboard approach of Hayes-Roth (1983) and Nii (1986a,b). The example domain used to test and refine the modeling tools is a remote sensor monitoring task, based on Naval Air Antisubmarine warfare. Despite its apparent specialization, this domain actually proves to be an ideal one in which to explore the research goals of the project and also possesses important features which allow generalization of the results.

The remaining parts of this introduction discuss the background to this research and introduce the Naval Air Antisubmarine warfare domain used in it. The following sections present the formalism developed to model RTMT human-computer interaction (section 2), the methodology used to acquire and model data of human computer interaction in the example domain (section 3), the details of the model of that domain (section 4 and the appendices), analysis and validation of the model (section 5), and conclusions from this phase of the research and implications for further research (section 6).

## 1.1 Toward Cognitive Engineering of Computer Human Interfaces

The transition of the underlying technology of person-machine systems from electromechanical to digital brought significant changes to engineering psychology in the 1970's and 1980's. The human operator was displaced from the traditional role of closed-loop control, into an outer control loop characterized by supervisory and decision-making functions (Johannsen,1976). The psychological implications of this new human role were clearly different from the more classical control-theory oriented engineering psychology exemplified by studies such as Fitts (1954) and Birmingham and Taylor (1954). The new human factors of computer systems became concerned with the human's ability to build effective plans, make insightful decisions, and solve difficult problems in the context of a given computer-human interface.

Attempts to build systems that dealt explicitly with specific outer-loop (decision-making) problems -- computer-based decision aids -- quickly showed that these systems had special needs (Schwartz and Jamar, 1983). They required a way of relating the structure of the computer interface to the structure of the task as understood by the human decision maker, i.e., some way of fitting the system to its user's cognitive processes. Faced with this need, engineering psychology has turned increasingly to cognitive science for new theories and methods.

This interest in cognitive human factors parallels the emergence of a consistent information processing theory of human cognition in recent years. (Pylyshyn, (1984) documents the history of the most widely held views, while Rumelhart, McClelland *et al.*, 1986, provide an overview of an alternative framework.) This theory postulates three kinds of constructs in explaining cognitive activity: mechanism, representation, and procedure. Mechanism refers to a basic set of information processing capabilities that are thought to underlie all more complex instances of cognition. In most cases, the cognitive mechanism is depicted as a computational device, albeit one with highly non-von Neuman properties. Representation refers to the information about the external world that is stored inside the human and manipulated by the information processing mechanism. The cognitive representation encompasses most aspects of the knowledge held by the individual. Procedure refers to the sequence of operations performed by the information processing mechanism on the representation to produce some cognitive activity. Procedure includes both procedural knowledge as well as architectural processing features of the underlying mechanism.

Most information processing theorists have viewed the basic mechanism for cognitive processes as some form of computation (symbol manipulation). This view is widely attributed to Chomsky, although Newell, Shaw, and Simon (1959) were actually the strongest early proponents of the view that the human information processing mechanism was computational. Much attention was devoted in the 1960's and 1970's to identifying the representations and procedures (i.e., the knowledge) involved in various cognitive activities, such as memory (e.g., Anderson & Bower, 1973; Quillian, 1968) problem solving (e.g., Newell & Simon, 1972), and planning (e.g., Hayes-Roth & Hayes-Roth, 1979; Sacerdoti, 1974).

The information processing models used in cognitive studies have proven to have a distinctly practical side via computer simulations of human information processing. Cognitive models that are procedural and computational can be programmed on a computer and used to solve pragmatic problems. Much of the recent work in 'expert

systems' for decision support is based on this approach (e.g., Buchanan & Shortliffe, 1984; Hayes-Roth, Waterman, & Lenat, 1983; Waterman, 1986). An information processing model of an expert decision maker is developed and programmed so as to act as a surrogate instead. More to the point for this study, computational models of computer users can be programmed and embedded within the user-computer interface as a way of giving the computer system a model of its user (Croft, Lefkowitz, Lesser, & Hufff, 1983; see also Norico & Stanley, 1989)

A key problem in using cognitive models in this way has been the lack of a general methodology to capture the plans and knowledge applied by users of computer-based systems. Such a methodology -- a cognitive task analysis language -- is needed to structure and guide the development of models of computer users, regardless of whether the purpose of these models is design analysis or actual incorporation as simulations into the interface itself. A cognitive engineering task analysis language for describing and analyzing the cognitive processes that give rise to human computer transactions could be used to answer the kinds of cognitive engineering questions faced during human-computer interface design.

Initial efforts to develop a cognitive engineering task analysis began with attempts to formalize and generalize the informal methods used to capture decision-making procedures in specific application efforts and basic research studies (Ericson & Simon, 1980, 1984). Subsequently, some methods for applying data derived from protocol analysis to system design have been developed (Rasmussen, 1986a, 1986b); and the need for, and benefits of, cognitive engineering in the design of any complex computer-based system have been recognized (e.g., Norman, 1986; Woods & Roth, 1988).

The first theory-based attempt to develop an analytic method to incorporate human information processing models into human-computer interface design was Moran's (1981) tool, the Command Language Grammar. Although it was a useful start, it was difficult to apply because of its excessive complexity. The real breakthrough came in Moran's later collaboration with Card and Newell (Card, Moran & Newell, 1983), in their book The Psychology of Human-Computer Interaction.

Card, Moran and Newell (1983) developed an elegant two-level model of human information processing in computer-human interaction, and a tool for analyzing it. The first (outer) layer of their model contains the procedure and representation part of the model: the problem-specific knowledge and cognitive processes used tc make a specific decision in a man-machine context. The second (inner) layer consists of the information processing mechanism used to execute the specified procedures. This mechanism layer was termed the Model Human Processor (MHP). The MHP consists of three inter-related and parallel information processing subsystems -- the perceptual, the cognitive, and the motor. Each subsystem has its own processing and memory capabilities. Performance parameters were defined for each subsystem component: cycle time for each processor, and storage capacity, decay rate, and code format for each local memory. Card, Moran, and Newell validated the MHP by quantifying the performance parameters through a combination of primary experimental data and results found in the literature. John *et al* (1985) and John and Newell (1986) have continued this work.

Card *et al* also developed a notation for describing information processing operations within the MHP architecture. This language was named GOMS after its four components: Goals, Operators, Methods, and Selection rules:

Goals: are states of the world that the person is trying to bring about. A goal may refer to a physical condition (e.g., "display data") or a cognitive condition (e.g., "understand cause of system failure"). Goals are hierarchically decomposed into lower level subgoals or direct actions, which may be either methods or operators. A goal with subgoals is accomplished by accomplishing all of the subgoals. A goal decomposed into only operators and methods is accomplished by performing all the operators and methods.

Operators: are elementary perceptual, cognitive, and motor actions which the person may undertake to achieve a goal or subgoal. This concept of an operator is not fixed but rather local to a specific level of analysis. At a very coarse level of analysis "open the door" may be an operator, while at a finer level of analysis "open the door" may be a subgoal and items such as "grasp the knob" and "turn the knob" may be operators. Operators are not unique to goals. That is, a given operator may be used as part of many different goals. This is of maximum importance in computer-human interfaces, where each command in the system represents a basic operator, and each command may be used in accomplishing many different goals.

Methods: are compositions of goal/subgoal/operator sequences into chunks. A method represents a partial or complete procedure for performing some subtask.

Selection Rules: are if-then rules for selecting among competing methods. In many situations the person knows multiple ways to accomplish a given goal, and chooses the appropriate way based on the specific instance at hand. This kind of control knowledge is encoded in selection rules.

GOMS was designed to be consistent with the architecture of the MHP and its empirical performance parameters.

The principle of hierarchical decomposition of goals in GOMS gives it great flexibility. Once a high level goal is defined, the primary conditions of its accomplishment can be defined as initial operators. Some or all of these operators can then be turned into subgoals and themselves decomposed. The variability in decomposition allows the model builder to focus on specific aspects of the cognitive process that are relevant to the research or design issue at hand. Of particular interest are likely to be those aspects that have observable or behavioral correlates, e.g., issuing commands, hitting keys, etc. This is ideal from a task analysis perspective, because it allows the analyst to begin at a coarse level of grain and proceed iteratively toward a desired finer level of grain in the task model, linking the behavioral correlates directly to the goal structure guiding their execution.

Card *et al.* (1983) have generalized a version of GOMS into an engineering-analysis model called the Keystroke Level Model (KLM) of human-computer interaction. KLM is used to translate a final GOMS model into performance-time predictions for a (new) interface through which the a task would be performed. Using the performance-time predictions, alternative interface designs can be evaluated. Kieras has extended the use of GOMS models for user interface design (Bovair, Kieras, & Polson, 1988; Kieras, 1988), deriving both quantitative and qualitative measures of cognitive complexity (e.g., estimating learning as well as performance times, evaluating interface consistency by whether similar goals are accomplished by

similar methods). A further use of GOMS models is building the interface around the procedural knowledge embodied in the model (Elkerton & Palmiter, 1989).

Much of the work done to date with GOMS by Card, Newell, John, Kieras, and others has focused on the lowest level goals and the operations embedded between them as a model of human-computer interaction. These researchers have all experimentally quantified detailed aspects of interaction at the keystroke level (e.g. performance time), using highly abstracted tasks or very simple tasks in which higher level problem-solving knowledge plays a small role. In the terms given above, they are interested in quantifying parameters of the information processing mechanism (modeled through the MHP) from data on human performance using a well-defined model of a specific procedure (represented as a GOMS sequence). By using simple or artificial tasks, they can, in essence, partition out the effect of problem representation and problem-domain knowledge on performance. This is an effective strategy for quantifying and/or validating the MHP model of human cognition, but not an effective one for interface design. That is obviously because in practice, human-computer interfaces will almost always address a task in which the user's knowledge is important. Whether the domain is computer programming or tactical ASW, the interaction between human and computer must take into account the representation and knowledge that the human has about the task.

This issue points out the two ways in which human-computer interaction (HCI) is being approached in current research. Cognitive psychologists and cognitive scientists, such as Newell, John, and colleagues, are using human-computer interaction as a vehicle to study and ultimately model human cognition. HCI is attractive for this task because it provides a rich yet tightly controllable context in which human problem solving can unfold and be experimentally observed. In this paradigm, any human-computer is an adequate source of data, and interfaces to 'toy' or synthetic tasks have special appeal because they allow the effects of individual expertise and experience to be eliminated. Human factors practitioners and engineering psychologists, on the other hand, are using cognitive analysis and theory as points of departure for the design of computer-human interfaces. In this paradigm, the task being performed by the person-machine system is of paramount importance, because, in general, the task is predefined by the application/design problem. The net difference is that cognitive researchers are interested in a cognitive task analysis language as a vehicle for modeling data on human cognition, while interface designers are interested in cognitive task analysis languages as a vehicle for identifying the task characteristics and user knowledge that they must address in the interface design.

The research reported below clearly falls into the second of these paradigms. It is primarily concerned with the use of cognitive models in designing new and effective interfaces for domains where all human operators are highly skilled, highly trained, and highly experienced experts. The role of task and domain knowledge in structuring and organizing the overall flow of human-computer interaction is therefore of central concern.

## 1.2 A Vehicle Tracking Problem Domain

It is almost impossible to study human cognitive processes (or tools to model them) without doing so in some specific problem domain. As discussed above, it was essential that this research deal with a task domain which was not synthetic but one in which real human experts existed. In addition, the domain had to require real-time problem solving on the part of the person, and to involve some competing demands for the person's attention.

The environment selected to do this was a vehicle tracking task, in which the tracking is done via remote sensing devices. This task is similar to the real-world domain of Naval Air Anti-Submarine Warfare (ASW), in which the vehicles being tracked are submarines, the tracker is located in an aircraft, and the sensors used to gather data are deployable on command from the human tracker. This problem in its most general form, has actually been the source of substantial study in both human-computer interaction (Goodson & Schmidt, 1989; Hopson & Zachary, 1982; Wohl et al., 1983), and in problem solving strategies (Durfee, Lesser and Corkill, 1989; Lesser and Corkill, 1981; Nii, 1986a,b). There are thus substantial results available in the literature on this problem. More important in this problem's choice, however, were its basic properties, which are described below.

Air Anti-Submarine Warfare (ASW) is concerned with the detection and identification of a hostile (i.e., target) submarine from an aircraft platform. The air ASW mission begins with searching an area of ocean where a submarine is thought to be and progressing through precise refinement of the target's location, tracking of the target (in peacetime), and destruction of the target (in wartime). This mission is performed by several cooperating crewmembers aboard a P-3 or S-3 aircraft. The crew typically consists of the aircraft pilot and navigator, one or more Sensor Operators (SENSOs), and other miscellaneous operators (not concerned with tactics). The Tactical Coordinator (TACCO) guides all of these personnel during the mission, coordinating their efforts and the available resources.

In the modern Air ASW world, virtually all information on the target is gained from a suite of sensors that includes passive acoustic sensors, or sonobuoys, active sonobuoys, RADAR, and Magnetic Anomaly Detection (MAD) sensors. Passive sonobuoys are the principal means of acoustic detection of the submarine. They are usually used in combination with active acoustic and nonacoustic sensors in order to achieve a sequence of increasingly refined information about the target. Below are the phases of a typical Air ASW mission:

1. Search -- The search phase is concerned with the initial detection of a submarine using passive sonobuoys dropped in the water to form a geometric pattern. SENSOs report contacts gained on particular sensors and the TACCO uses tactics to gather further information by dropping additional passive acoustic sensors or by the use of active sonobuoys or nonacoustic sensors.
2. Direct-Path Contact -- Occurs when the submarine is detected by being within close range of a sonobuoy. When a target is detected, the sonobuoy provides information about the target's direction (bearing). At this point, the TACCO often drops additional sonobuoys in order to get a cross bearing. This phase is essential to further localization.

3. Target Fix ---- When the target is in the direct path of the sonobuoy and the bearing information is available, the TACCO often drops additional sonobuoys in order to get cross bearing information. If the new sonobuoys gain contact and the directional information of two buoys intersect, this provides a locational hypotheses (target fix).
4. Target Track ---- After one fix is obtained, further fixes are sought after with the use of acoustic or nonacoustic sensors. Any subsequent fix permits determination of target course and speed. Further tracking occurs with the deployment of additional sensors and/or the use of nonacoustic sensors.
5. Attack Criteria ---- This phase requires fulfilling a specific set of requirements defining the fixing/tracking accuracy necessary to conduct an effective attack. This involves achieving a particular level of correlation across a number of different sensors.
6. Attack/Kill ---- Deploying a weapon compensating for the inferred target location, and the differential movement of the aircraft, target, and weapon. This phase also involves preparing for a second attack if it is necessary to do so.
7. Alerted Target ---- When a submarine detects the presence of a threat from an ASW aircraft, it will take evasive action immediately, making it much more difficult to pursue.

During the search phase, a physical phenomenon called the convergence zone, or CZ, phenomenon often occurs and makes the use of passive sonobuoys much more difficult. An acoustic sonobuoy can 'hear' sound directly propagated from an emitting source over a small distance; this is called its direct path (DP) detection range (e.g., 2 - 5 nautical miles). Because of ducting of sound underwater, there may be a small annular region quite distant from the DP zone in which detection may also occur. This is called a CZ. There may be none, one, or possibly two CZs in any given acoustical environment. The presence of CZs creates a complex pattern of potential detection regions in a field of sonobuoys. Direction passive sonobuoys also provide a bearing to the target, but this bearing contains error, and does not help disambiguate whether the sound originates from the DP zone or from a first or second CZ.

Thus, the heart of the Air ASW problem involves using this ambiguous, errorful data from fields of sonobuoys in conjunction with data from active sonobuoys and nonacoustic sensors to detect the presence of a submarine and iteratively refine its location, course, and speed. The problem requires the TACCO to continuously revise target hypotheses based on the current situation and plan new tactics to gather further data, which in turn will cause an update of the hypotheses. This process an carried through repeatedly until attack criteria are gained, which entails having a pre-specified degree of certainty about the target's location course, and speed.

Embedded within this is that fact that the TACCO must direct the aircraft to deploy new sensors as a way of disambiguating data from existing sensors. This makes the movement dynamics and attendant time-lags another part of the decision process. Often, for example, the TACCO may know where to place an additional sensor, but can not get the aircraft to the desired location in time for the data to be meaningful.

## 1.3 Rationale for Domain Selection

Within the Air ASW domain, the individual responsible for integrating the information from the aircraft's sensors and for locating and tracking a vehicle, the TACCO, was chosen for study. The TACCO manages the resources and the other personnel aboard the aircraft to achieve the goal of the mission (i.e., to locate the submarine). The TACCO's methods for achieving this goal constitutes a RTMT problem domain and was chosen for this study for the following reasons:

1. There are pre-existing human experts available for use in experimental procedures to gather data on human-computer interaction and problem solving. This provides a great benefit of an Air ASW problem over a purely synthetic domain. In the synthetic domain, the experimental participants must first be trained in the task(s) involved. Time and budget constraints often lead to the design of highly simplified tasks to avoid in-depth training of participants. However, complexity is a primary attribute of RTMT domains. Basing research on an existing domain provides the complexity needed without requiring a lengthy training period for the participants.
2. Virtually all problem-solving in real Air ASW occurs through the on-board computer; thus, the operators are already familiar with computer-based problem solving and their current strategies are built around a computer interface. Another consequence of this pre-existing computerization is that all operator and system actions are implicitly 'available' for study and modeling via on-line data recording.
3. The nature of the computer interface to this task is graphical, thus eliminating any aspects of natural language from the problem-solving process. The graphical 'language' of the interface displays all information as graphical objects on the screen, and requires all operator actions as direct manipulation of those objects. For example, the area which is being searched for hidden vehicles is depicted as a flat map, and the system user causes a sensor to placed into this environment by pointing to a map location and invoking a function to place a sensor at the selected location. This generates a steering command to the pilot (shown as a steering symbol on the map/screen), and a set of commands to the aircraft computer to drop a sensor when the pilot arrives at or 'captures' the desired map point.
4. The underlying problem environment is known to be simulatable: the movement of the aircraft, the action of acoustic sensors, and many other important aspects of ASW can be simulated to a reasonable degree of realism with standard simulation tools. This allows the creation of a realistic workstation for the user, but one whose problem dynamics are driven by an embedded simulation program. The human TACCO could thus have the appearance of working at a realistic interface, but in a digitally based and tightly controlled environment.
5. Human-computer interaction (HCI) problems have long been reported in this domain, and it can thus benefit from enhanced human-computer interaction techniques such as are the ultimate goal of this research.

Aside from these obvious advantages of the ASW domain for experimentation, several characteristics of the domain also make the methodologies and tools used to model it generalizable to other domains:

1. Non-critical real time -- While ASW is a real-time problem solving environment, it is not time critical. Events unfold not in terms of seconds, but rather in terms of minutes or fractions thereof. Events unfold relatively slowly as compared to domains such as aircraft maneuvering, where the operator (pilot) must react within seconds to changing situations. This time scale is more comparable to domains such as factory control, power-plant management, etc.
2. Multi-Tasking -- The operator must always be ready to redirect attention to differing tasks based upon the unfolding situation and the priorities attached for the tasks at any given time. For instance, if the individual is performing a routine task of environmental assessment and then receives some unexpected sensor contact, the operation must suspend what is currently being done and turn attention over to investigating the contact. This is the basic property of all of the multi-tasking domains listed at the start of this paper.
3. Opportunistic -- Because it is often based on unpredictable events, the operator's attention switching and decisions to perform certain actions may be opportunistic depending on current hypotheses about the overall situation and available resources.
4. Data integration -- The operator must continuously integrate data from a number of different sources, such as acoustic data, environmental data, and existing mission constraints. The individual uses this data to revise existing hypotheses about target behavior, the environment, and other situational factors.

All of these characteristics are present in domains such as nuclear process control, command and control, and air traffic control; where the operators must integrate data from a number of different sources and perform tasks opportunistically depending on a situation which is slowly unfolding. The methodologies and tools used in this paper were developed to deal with all of these characteristics for the ASW domain. Thus, these tools could be applied to many other similar domains for the development of automated intelligence.

## 2. FORMALISM FOR COGNITIVE DESCRIPTION OF REAL TIME MULTI-TASKING HCI

The use of abstracted tasks as a vehicle to study human behavior has long been the dominant paradigm in experimental psychology. Human factors or engineering psychology has tended to follow this paradigm, using laboratory based observations and/or task model and then extrapolating them to operational settings. In the 1970's however, cognitive science began to offer another paradigm, based on in-situ analysis of human beings performing complex, real-world tasks. This new approach was motivated by a desire to model human expertise in these domains and to simulate this expertise in computer-based expert systems. In addition to focusing research attention on the role of experience and knowledge in human performance, it has more generally left the door open for reconsideration of domains where the simple extrapolation of laboratory or abstract task data to real-world issues may be inadequate. The case of real-time, multi-tasking (RTMT) environments appears to be such a case.

Real-time multi-tasking occurs in domains such as:

- vehicle operation,
- air or surface vehicle traffic control,
- plant/factory operation,
- telecommunication network or remote sensor monitoring,
- military command an control.

These domains are far removed from the abstracted tasks most often studied in the laboratory. First and foremost, typical users are highly trained and practiced, and in fact require substantial training and practice (on the order of hundreds or thousands of hours) to achieve acceptable levels of performance. Thus, these are similar to the 'expert' domains studied under the expert systems rubric in that experience and expertise play critical roles in human performance.

RTMT domains have other constraining features in addition. They are generally what Hewitt (1985) referred to as open problems. Because they occur in natural , uncontrolled situations, the domains do not have clear boundaries or at least do not operate near a clear set of boundary conditions. In RTMT domains, both spatial and temporal dimensions assume a greater importance. The temporal dimension is obvious, as system users must adapt their behavior to the real-time pace of events in the underlying problem as well as to the real-time lags that can occur between control inputs and the results of those inputs. A major implication of this is that humans in RTMT environments must form expectations about possible future events, and factor those into the way that they organize their own behavior. The importance of the spatial dimension is more empirical. In the general areas where RTMT occurs, the problems tend to have a spatial component (e.g., vehicle operation, air or surface vehicle traffic control, plant/factory operation, telecommunication or sensor monitoring, military command an control are all domains that have strong spatial aspects to the problem).

These spatial aspects require the human system operator to perform extensive spatial interpretation and reasoning steps throughout each problem.

Another common attribute of RTMT domains is the presence of high levels of uncertainty about present and future events. Many of these domains (including all those listed in the preceding paragraph) involve other agents or systems that are separately controlled. This creates the opportunity for unexpected behaviors, and hence uncertainty about the evolution of events. Most RTMT domains involve physical or mechanical processes whose dynamics are extremely complex and often incompletely understood. In addition military domains involve agents that are acting in a hostile or adversary fashion, and may be deliberately trying to deceive the human system operator. Finally, the RTMT operator interacts with the environment through the workstation itself, which then electronically controls the physical subsystems involved. This not only creates time delays, it also creates uncertainties as to the result or outcome of actions. In the nuclear plant operation, the operator may give a command to close a valve, but may never be certain whether the valve really closed completely , because the closure can not be physically observed.

## 2.1 Requirements for an RTMT Model of HCI

These attributes of RTMT environments, together with the goals of this research outlined in Section 1 above, give rise to several requirements for a formal model of RTMT human-computer interaction. These requirements fall into three classes: psychological, computational, and operational.

One psychological requirement is that the formal model deal both with the behavioral or observable aspects of human-computer interaction and with the underlying (and unobservable) cognitive processes that give rise to the observed behavior. Most existing techniques focus on just one of these aspects. Human factors methods such as classical task analysis models such as HOS (Glenn, 1988), MicroSaint (Chubb, Laughery and Pritsker, 1987) typically consider only behavioral aspects. A purely behavioral approach can not provide insight, however, into how the context of the individual problem affects the problem solving approach and thus the pattern of observed interaction. This requires a cognitive focus, such as provided by models such as Rasmussen's (1986) decision ladder or Newell and Simon's (1972) problem behavior graph. These approaches, however, do not explicitly consider the observable behavior of a human operator in sufficient detail to allow an intelligent human-computer interface to interpret operator goals or intentions from detailed observations of sequences of actions. (This is a major application goal of this research). The GOMS notation of Card, Moran and Newell (1983) is an example of an approach that provides both a cognitive and behavioral frame of reference, since a GOMS model identifies a sequence of observable actions that are expected in a given problem context, and also indicates how these actions are related to the intentions and goal structure of the person.

A second psychological requirement is that the model be descriptive rather than normative. It should not specify how the problem should be solved or optimized, but rather what a human operator would do in a given RTMT problem situation. A descriptive model is necessary to allow the model to be translated into a user model

(see Norman, 1983) that would allow a computer interface to reason about and interpret the user's actions. The 'extreme' descriptive position can be avoided by eliminating consideration of performance errors on the part of the person, and instead focusing on describing what is sometimes called the operator's competence. This sort of model describes a person's knowledge about how to solve a problem or perform a task without addressing how the application of that knowledge might be affected by cognitive limitations (such as short-term memory span or the pace of real-world events) or physical limitations and errors.

A third psychological requirement is that the model explicitly treat both the real-time and the multi-tasking issues. That is, the model must provide ways for the constraints of real-time systems to affect the way in which the problem is solved, just as it must explicitly treat the ways in which the computer user will adjust his/her attention among tasks to compensate for the real-time dynamics and evolution of the problem. Finally, the model must allow for both the specific problem instance and its evolution (i.e., the local context) and the experience of the human operator to affect the way in which the human-computer interaction proceeds.

The main computational requirement for a RTMT model of HCI is that the model must be computable. Computable here refers to the ability to represent the model unambiguously as a series of computations either on an abstract device (e.g., finite state automaton) or a real computer. Computablity is important for two reasons. First, a computable model can be translated into a cognitive simulation which can then be used to explore, test, and/or even disprove the model itself. Many models of cognitive processes are conceptual rather than computable (e.g., Woods and Hollnagel's, 1987, *goals/means network) and thus can not be subjected to this form of empirical analysis.* Second, a major motivation for developing this class of RTMT models is to use them as embedded users' models in human-computer interfaces. This is not possible if the representation used in uncomputable.

Finally, the model must be capable of incorporating the operational constraints of real-world environments. As noted above, RTMT problems occur commonly in work contexts, despite having received little study in laboratory experiments. It can be argued that there is no meaningful abstract analog of RTMT domains because of the important empirical role that operator expertise and experience play in them. Thus, any RTMT model or modeling framework must be flexible and open enough to adapt to a wide range of realistic constraints and features of problems that are likely to be encountered in application domains.

## 2.2 Framework for Representing RTMT HCI

One of the earliest conceptual models of real-time multi-tasking cognitive processing can be found in the early work of Selfridge (1959). He proposed a "pandemonium" metaphor of cognitive processes composed of "shrieking demons." Each demon was able to perform some aspect of cognition and shrieked for attention as an opportunity arose for that process to occur. As the situation became closer to the ideal conditions for the demon, it shrieked louder and louder. Attention, in Selfridge's model, was simply the process of placating the shrieking demons by allowing the loudest one to act. Real-time multi-tasking arises in this conceptual framework when

the context and temporal dynamics of the problem allow (or require) many of these shrieking demons to compete for attention in such a way that no one of them maintains control for very long. This shrieking demon metaphor provided the impetus for a great deal of work in problem solving, including the original work on blackboards and opportunistic control structures (c.f. Nii, 1986a), and on spontaneous computation components in cognitive simulations.

The framework used in this study for representing real-time multi-tasking aspects of HCI is quite similar to Selfridge's notion. Our framework conceptualizes the person as a network of activities, each of which represents a partial or local strategy for performing some task or for solving some aspect of the overall problem. The flow of attention from one task to another is triggered by momentary changes in the problem environment, which may be the result of actions taken by the person or the result of actions of other agents and/or the environment. It is important to note that, in the HCI case, changes in the problem environment can be equated to contextual changes on the workstation screen, since the screen is the users' window into the problem environment. Figure 2-1 shows an abstraction of how attention flows through this network of tasks.

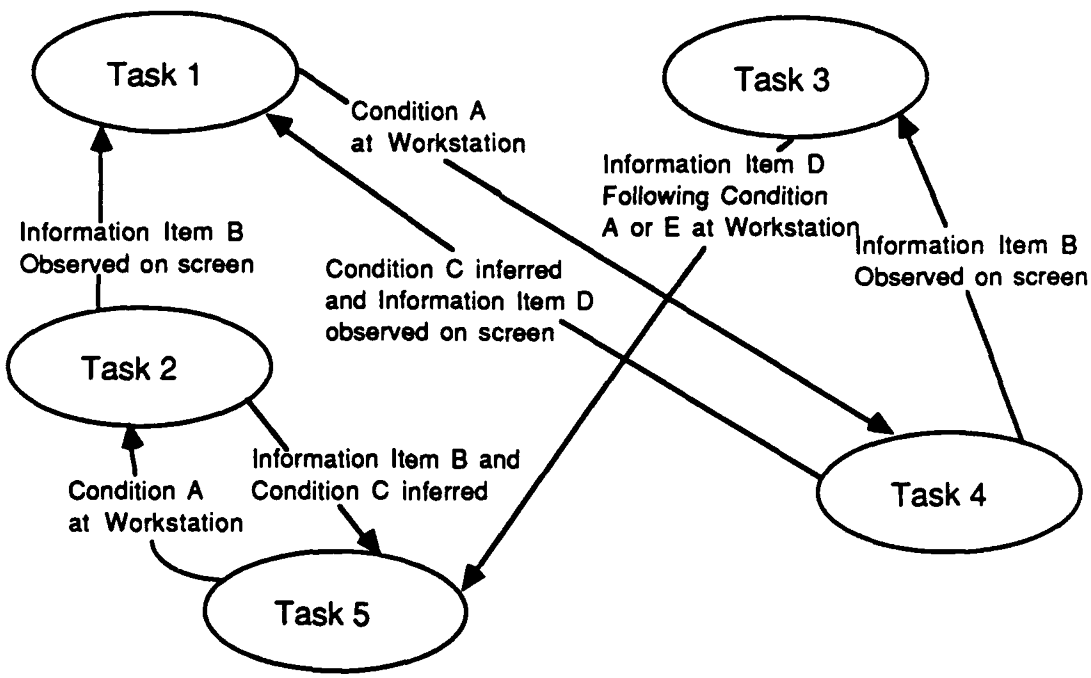


**Figure 2-1 COGNET Concept**

In Figure 2-1, the user is performing a task ("Task 1"), when he/she receives notice of some condition at the workstation, such as an error or warning alert. This explicit condition triggers the user to defer further work on Task 1 and instead to initiate work on Task 4 (perhaps a trouble shooting task). While performing Task 4, however, a piece of information is observed on the screen. This datum, in the context of the trouble-shooting task and the user's prior experience with this specific information item, causes her/him to suspend Task 4 and instead begin Task 3 (perhaps an

analytical or information gathering task). The analysis performed in doing this task then uncovers yet another piece of data ("Item D"), which in the context of the original condition (Condition A in Figure 2-1) leads the user to cease the analytical task (Task 3) and initiate yet another task (e.g., Task 5). While performing this task, however, the user encounters condition A again (as he/she did while previously performing Task 1). Because the context is different now, though, the response is also different. Whereas previously Condition A triggered an initiation of Task 4, in this case it triggers an initiation of Task 2. Figure 2-1 thus indicates various factors that can influence shifts in attention among tasks -- explicit cues, prior knowledge, local problem-solving context, and associations or inferences (i.e., knowledge).

These various kinds of shifts of attention among tasks correspond to different control principles that have been discussed in the literature on symbolic computation. There are basically only three general notions of control in symbolic computation systems:

- forward or data-directed, where the choice of "what to do next" proceeds forward systematically from the initial condition or initial data toward some goal or solution state;
- backward or goal-directed, where the choice of "what to do next" proceeds backward systematically from the goal state or solution back toward the initial condition or data; and
- opportunistic, where the choice of "what to do next" proceeds on the basis of what is the most appropriate action at that point in the problem-solving process, regardless of whether the immediate action moves forward from data or a partial solution or backward from a goal toward data or initial conditions.

The attention shifts that represent responses to unexpected data or information are data-directed shifts. Those that represent deliberate invocations of pre-existing methods or procedures are goal-directed shifts. And those that are based on associations or knowledge gained through experience applied in unanticipated ways to the local situational context are opportunistic shifts.

We have called the conceptual framework in Figure 2-1 the COGNET (for COGnitive NEtworks of Tasks) model. One element of COGNET not anticipated in Selfridge's metaphor is the basis for the coordination among the various tasks or demons. Coordination, as defined in Malone (1989), refers to the means by which cooperating but independent agents organize their individual problem-solving activity to achieve a solution to a more global problem; the study of these mechanisms for coordination and cooperation is termed 'coordination theory.' While coordination theory (see Robertson, Zachary and Black, 1989) usually presumes a set of completely independent agents operating in parallel, it is still highly relevant to this COGNET framework which deals with only a single individual. In COGNET, each task acts as an independent problem-solving agent (like one of Selfridge's demons). Each task may be partly completed, interrupted by some other task, and perhaps later resumed at the place where it was interrupted. The tasks are thus all in various states of completion at any one time, giving them the appearance of concurrence, albeit not actual simultaneous activity. We consider this to be a form of weak concurrence (as

opposed to the strong concurrence of completely independent, active-in-parallel agents). Moreover, these weakly concurrent tasks (unlike Selfridge's demons), are also all interrelated, in that they each contribute to some higher-level problem-solving goal. This common interrelationship among the tasks -- their linkages to a common goal -- implies some mechanism for coordination among the tasks.

Coordination theory suggests two general mechanisms for instituting coordination among the various tasks in a COGNET network -- either explicitly or implicitly. Explicit control requires some executive or controlling task or process. This is a more straightforward approach, but also gives rise to a large set of theoretical difficulties and paradoxes that led to the discarding of the notion of an explicit 'cognitive control' process back in the 1970s. Implicit control, on the other hand, requires a mechanism by which the behavior of each task is somehow constrained in a way that leads toward the solution of the overall goal. This is the approach used in COGNET. Coordination among the weakly concurrent tasks is established through their use of a common global problem representation. That is, all tasks use the same (overall) representation of the problem being solved. Each contributes to some specific portion of the problem (as bounded by the representation), acting to move that portion of the representation toward a solution (or at least away from some losing or unacceptable state). This approach is preferred for two reasons. First, it avoids the difficult theoretical problems of a control 'homunculus' that an explicit control process would required. Second, it best fits various anecdotal and formal thinking-aloud data (including that reported below in Section 3) that indicate in real-world tasks expert problem solvers used an integrated problem representation, although actual problem solution requires a multi-tasking approach (see also Zachary, 1989).

The COGNET framework is actually a meta-model, or architecture, for building models of specific RTMT domains. The flow of cognitive processing in a COGNET model resides at any given moment in a specific task in the network. The focus of attention remains there for some time until it is captured by another task/decision node (by a change in the problem representation that enables the second task node to capture control from the first). The flow may also be opportunistic, when a goal-based shift is enabled by a specific state of knowledge in the problem representation. Overall, the flow of attention between and among the tasks both reflects the changing context of the problem evolution (via the changes in the common problem representation) and also contributes to the problem evolution and change in context by directing the specific sequence of operations that are performed by the sequence of tasks that gain attention and are performed.

The COGNET architecture itself meets some of the requirements for an RTMT representation listed above. It represents real-time multi-tasking in such a way that performance is sensitive to both situation effects and the experience and knowledge of the human operator. Experience and knowledge affect both the methods encoded in the individual tasks in the network as well as the triggers that allow for attention flow between tasks in the network. Situational effects are also effected by the attention mechanism, as well as by the use of a common, problem-specific, representation by all the tasks in the network. The other required attributes of the RTMT representation depend on the formalisms used to build specific RTMT models within the COGNET architecture.

## 2.3 Formalisms for Building COGNET Models

The full COGNET framework consists of two separate but interrelated pieces. The first is a particular set of task models which the user follows to solve the RTMT problem involved, and the second is an integrated problem representation that these individual tasks use in their own procedures and that , in addition, constrains their interaction into a coordinated problem-solving process. The central component in a COGNET model is the problem representation. As a formal entity, it requires four properties. It must be:

- sharable by the separate, weakly concurrent activities that make up the tasks in the COGNET network;
- dynamic, capturing changes both in the problem environment and in the state of the problem solution;
- constructive, acting as the medium by which a solution to the problem is built over time by the aggregate results of the separate tasks in the COGNET network; and
- flexible enough to allow goal-directed, data-directed, and opportunistic attention shifts.

Interacting with this component, but not directly with each other, are a series of independent tasks. Each task is a unit of local, goal-directed, procedural knowledge about some problem-solving activity that contributes to the problem solution. The task is local in that even when successfully completed, it can not by itself solve the overall problem. The task is goal directed in that it is defined by a specific goal that indicates the local function of the task in the overall process. The task embodies procedural knowledge in that it exists as a set of (goal-directed) operations that, when executed, may achieve the (local) goal that defines the task. Each task, when active, processes information on the common representation using inferential tools (i.e. cognitive operations) and manipulative tools. The latter are actions that the person can take through the human-computer interface to affect change in the real-world or to manipulate information in it. Each task can and does change the contents of the common representation according to the operations performed during the task. These changes can affect other tasks indirectly through the common problem representation, but tasks never interact directly.

The next sections describe the specific tools used to formalize these two components and build a specific COGNET model.

### 2.3.1 Modeling the Global Problem Representation

The technique used to formalize the problem representation in a COGNET model must able to support the (weakly) concurrent problem-solving flavor of COGNET, and have the properties indicated above. The best developed structure for this (and arguably the only one) is the blackboard framework. It is based on a simple scheme, first put forth by Newell (1962), of multiple problem-solving agents achieving mutual control by working from a common data structure on which all initial data, partial

solutions, and goal states are maintained. Each agent is able to perform only certain kinds of tasks on certain kinds of data, but is able to recognize when the conditions exist for its task to be performed. When applied to COGNET, these agents are equated to the weakly concurrent tasks in the cognitive network.

The blackboard acts as a global data structure which can be independently examined by each task when it is active, and changed by that task as part of its activity. However, each time there is a change to the blackboard, each other (inactive) task is also able to examine the blackboard's (new) contents and determine whether the conditions now exist for the agent to become active. In Selfridge's shrieking demons metaphor, each change to the blackboard allows each task (i.e., demon) to re-evaluate how loud it is allowed to shriek. If and when it becomes loud enough, it activates itself and performs its task, which will result in some modification to the blackboard contents that enable some other task to seize control. If not, the task simply waits until the next change to the blackboard. Thus, each task operates opportunistically, with no additional explicit attention mechanism needed.

The first major application of the blackboard approach was in the speech understanding system HEARSAY-II (see Erman et al, 1980). Recent articles by Nii (1986a,b) review the development of blackboard systems from the original HEARSAY approach forward to the present. Nii (1986a) also provides a general formalization of the blackboard concept. In recent years, Hayes-Roth, et.al. (1983) have applied the blackboard framework to development of cognitive models, and proposed it as a very general framework for problem solving. As Nii (1986a) notes, there is great flexibility in applying the blackboard concept to specific domains. There are, nonetheless, several principles that define the concept:

1) hierarchical layering of the blackboard: the blackboard is divided into layers which contain information at hierarchical levels of abstraction, with the solution level at the top and input data or initial conditions at the bottom. Each intermediate layer may contain a distinct vocabulary and syntax, but the relationships between layers must be well-defined (e.g., part-of, instance-of, etc.). The blackboard may have multiple independent panels with separate hierarchies.
2) independence of agents: each agent is fully compartmentalized from the others. It acts as if it were the only agent, and recognizes conditions for its activation only by examining the blackboard contents.
3) globality of the blackboard: the blackboard is available to all agents, and can be modified only through the actions of an agent.

It is possible for the blackboard to be separated into several independent pieces or panels. These panels can reflect different aspects or viewpoints on the problem domain. In the COGNET application to Naval Air ASW discussed in Section 4 below, the blackboard has two separate panels, one reflecting the view of object being sensed (the target panel), and another reflecting the relationships among all domain components (the situation panel).

It is important to note how the COGNET framework differs from a 'pure' blackboard structure. In general, the 'tasks' in COGNET are more complex than the simple

knowledge sources in the pure blackboard framework, and they have some elements of control incorporated within them. They represent compiled 'chunks' of problem solving activity, just as intermediate blackboard levels represent constructed 'chunks' of an overall solution. In a pure blackboard structure, such as in systems like HEARSAY-II, the problem solving process is broken into small steps that correspond to application of individual knowledge sources (see Erman, et al, 1980). Each knowledge source can recognize a specific situation on the blackboard and make a specific change to the blackboard. Usually this change amounts to posting a new piece of information at the next higher or lower blackboard level, depending on whether the knowledge source is expressed in a forward directed or backward directed manner (respectively). The knowledge sources are narrowly focused (on single conditions and single actions), so there are potentially a large number of them. All control processes are externalized from the knowledge sources, in keeping with their simple structure. The flow of attention within this formalization of the COGNET architecture is shown in Figure 2-2. A given Task (e.g., Task B in Figure 2-2) may be active at a certain point in the problem evolution Task B 'reads' or uses certain information from the current blackboard contents in its task processes, and may then subrogate to Task A to provide more localized or complementary analysis. Task A both 'reads' information from the blackboard and posts new information to the blackboard. This new information, upon its posting, may then complete a blackboard pattern that satisfies a triggering condition for Task C, which then captures control. Task C similarly posts new information on the blackboard, but this does not satisfy the pattern associated with any trigger with sufficient priority to displace it. Ultimately, however, the activity undertaken in Task C leads to an action which involves a substantial time delay before its effects are known. Task C then suspends itself until these effects are known (i.e., posted on the blackboard), and this allows Task D to capture control. Task D previously had its triggering condition satisfied but had insufficient priority to capture attention from Task C, which is now suspended.

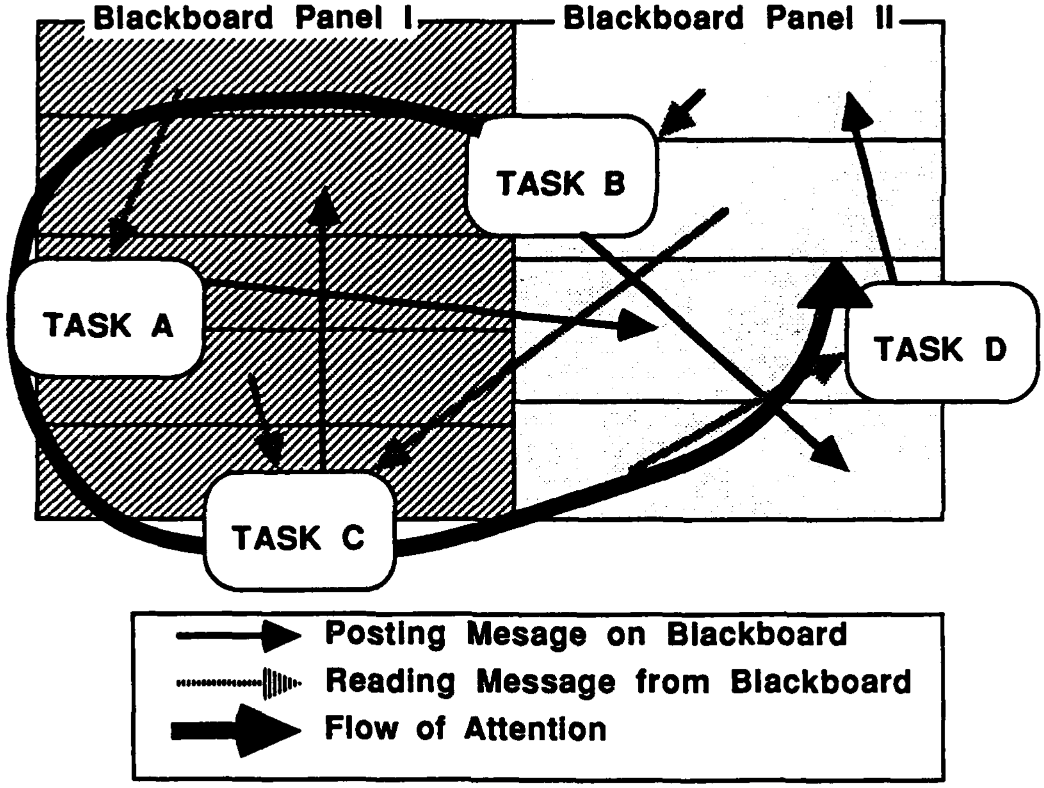


**Figure 2-2 Attention Flow Between Tasks**

2.3.2 Multi-Tasking and Attention Flow in COGNET

The tasks that make up the COGNET network represent larger chunks of procedural expertise. Each COGNET task comprises a relatively self-contained procedure for accomplishing some activity or developing some partial problem solution. Thus, unlike the 'pure' blackboard knowledge sources which are narrowly focused, each COGNET task is very broadly focused, in that from initiation to completion it may make several changes to the blackboard. This difference has other broad implications for the flow of control or attention.

In pure blackboard models, each agent or knowledge source performs only one action, and hence is indivisible. Attention is reallocated after (and only after) each knowledge source is finished performing its operation. Thus, control is only directed between knowledge sources. In COGNET, there is an analogous attention process which determines the shift in attention from task to task, but there is also a control process within each task. The flow of control within each task is forward directed, because each task is defined by a single goal and consists of a method for accomplishing that goal. In fact, during the majority of the duration of a typical RTMT problem, the person will be involved in applying one or another of the goal-directed procedures (i.e., COGNET tasks). Thus, most of the flow of cognitive processing does not involve attention shifting.

In addition to the goal-directed attention flow within each task, there are three different ways in which attention can flow between tasks in the blackboard-based COGNET network. The first is explicit capturing of control, in which one task spontaneously assumes control from another in response to some change in the problem state. This capturing of attention is activated by the occurrence of some triggering set of conditions in the problem or current state of the solution. Such triggering conditions are formalized as patterns of information on the blackboard. When a set of information posted on the blackboard satisfies some triggering pattern, attention flows to the task associated with the trigger, and the task from which control was captured becomes inactive. The interrupted task may or may not be able to recapture control and complete execution at some later time, awaiting some future pattern of information that may allow it to recapture attention. It may frequently be the case that more than one triggering pattern may be satisfied by the same blackboard contents. In such cases, multiple tasks may be vying to capture attention from the currently active task, and an explicit priority must be assigned to each trigger to establish a precedence order among them. These priorities are reminiscent of the loudness of the shrieks of Selfridge's various demons.

The second type of attention flow occurs when one task suspends control. This is particularly important in real-time problem domains where there is often a delay between the time some operation is activated at the workstation and the time its effect is known, for example between the deployment of a sensor and the time it gains a contact. Along with the suspension, a local or temporary trigger is created, which will allow the suspended task to recapture attention at some point in the future. Typically, this temporary trigger is the expected result of the action taken prior to the suspension, or perhaps a 'time out' event (defining a future time at which the absence of an expected result will be taken as an indicator of an unsuccessfully completed action). In either case, once this temporary trigger pattern is satisfied by the blackboard contents, the suspended task will again begin to compete for attention.

Unlike the attention capturing case, there is no clear choice for which task can assume control when control is suspended. Instead, all of the inactive tasks vie for attention (like so many shrieking demons) as they always do, but the task which has suspended itself no longer has attention or is competing for it. This leads to one of two cases. In the simplest case, there is another (second) task whose triggering pattern is satisfied by the current blackboard contents but which simply had less priority than the previously active (i.e., first) task. Now, with the first task suspended, the second one is able to capture attention. This case is presumed to be the case that occurs, because most RTMT domains seem to have one or more general background tasks associated with overall monitoring or system housekeeping that can be invoked at any time, but only have the priority to do so when there is little other task activity.

The second case occurs when there is no other task whose triggering condition is met by the current blackboard contents. In this case, the person simply waits until a change in problem conditions allows the suspended task or some other to regain attention. For completeness, it is noted that a special case of suspension is when a task is fully completed and terminates execution.

The third way in which attention can flow is termed subrogation of control, in which one task suspends control but passes it directly to another task. This subrogation may also involve establishment of a temporary or local trigger for the subrogating task (viz.,

that control will return to the original task when the second task is completed). Like all triggers, however, this does not guarantee that the subrogating task will ever eventually regain attention, merely that it will reach a point where it can again compete for it.

In general, capturing of attention corresponds to those cases where the user might say that a 'red flag was raised' or a 'mental alarm went off.' Suspension, on the other hand, corresponds to cases where the user might simply have noted the need to 'stop task A and move over and do task X, while waiting for event Y to happen.' Finally, subrogation corresponds to those cases where the user deliberately decides to seek further information, undertake more detailed analysis, etc., before completing the current task.

### 2.3.3 Modeling Individual Task Performance

The procedural aspect to a COGNET model is entirely encapsulated in the individual task components of the COGNET network. The panel and layer structure of a COGNET blackboard is defined by the COGNET model-builder. This blackboard structure, in any specific COGNET model, acts as an integrating but completely passive component. All human-computer interaction, as well as all posting, unposting and transforming of information on the blackboard is the result of application of procedures that make up the individual tasks. To this end, a detailed cognitive task analysis language has been developed and incorporated into COGNET to represent the procedural knowledge that is contained within each task in a COGNET model.

The basis for this COGNET task language is the GOMS notation developed by Card, Moran and Newell (1983) and introduced above in Section 1. GOMS provides the basic goal-directed flow of cognitive processing that is characteristic of the individual tasks in COGNET. A GOMS model consists of a single high-level goal, which is decomposed into a sequence of one or more subgoals and/or operators. At any given level (including the highest level), the processing proceeds sequentially through the subgoals and operators at that level. A subgoal is considered to be 'met' if either:

- the conditions for its pursual are not satisfied (i.e., the goal is locally irrelevant) or
- the goal is locally relevant and all operators in its sequence are applied, and all subsubgoals within the sequence are also met.

These two conditions really define a GOMS model as goal tree, in which goals are pursued in a depth first manner.

The original GOMS notation provides a convenient starting place in creating a COGNET task description language, for several reasons:

1) because the applicability of any lowest level operator is defined by the highest level goal being pursued, GOMS has the desired goal-directed control aspect for COGNET task models;

2) GOMS allows cognitive and motor operators to be freely intermixed in a given sequence (or method) to accomplish any goal or subgoal, making it appropriate for the dual cognitive/behavioral description requirement for COGNET;

3) GOMS allows conditions to be placed on the applicability or relevance of individual goals or operators, thus making it (at least structurally sensitive) to problem context; and finally,

4) the basic notation of GOMS follows a fixed syntax, making GOMS representations computable.

There were also several limitations that required substantial enhancements tc GOMS for the COGNET task description formalism. First, GOMS was completely tied to goal-directed processing. There was no way to link GOMS models into a COGNET-type network, nor to allow them to capture attention from one other, suspend their own processing, or subrogate to other tasks. Second, there was no structure included in GOMS to tie it to any formal problem representation. GOMS was and is a completely procedural formalism, incapable of constraining its flow of operations on the basis of an external or mental model/representation of the problem domain. Third, there is no representation of time in GOMS, and as a result no way to deal with expectations of future events or reasoning about time, as is required in real-time domains. And fourth, there has never been any clear agreement as to the desired level of detail in a GOMS description of a task. Card, Moran and Newell (1983) originally used several levels of detail ranging from functional to keystroke level. Their 'engineering' version of GOMS, the Keystroke Level Model(KLM) focuses exclusively on individual keystrokes, but also loses most of the ability to deal with cognitive operations. Recent research by John (see John and Rosenbloom, 1985, or John et al., 1986) also focus on the keystroke level in validating the MHP on which GOMS was built, but others, notably Kieras (1988) and Elkerton et al. (1989) emphasize the functional or part task level of analysis. For COGNET, a single and consistent level of grain must be specified.

To deal with these limitations, a substantially different cognitive task analysis language was developed from the original GOMS framework of Card et al. This full COGNET task notation is shown in Figure 2-3. The basic goal/subgoal/operator structure of GOMS is maintained. However, each task is defined by a single top-level goal, which has associated with it one or more triggers. These are formal descriptions of patterns of information that, when observed on the problem blackboard, allow this task to capture control from other tasks. As an optional feature, a numerical priority is assigned to each trigger, such that no two triggers (regardless of the task they trigger) have the same priority. This avoids the problem of trigger deadlock.

GOAL: GOAL NAME...*Trigger*

GOAL: SUBGOAL NAME...<...*conditions*>

OPERATORs <...*conditions*>

Perform FUNCTION. <(accompanying data/ parameters)>
where FUNCTION=any invokable function
Point element/location on screen
Enter alphanumeric data in response to cues
Select item from screen
Post object on blackboard
Unpost object on blackboard
Transform object on blackboard
Suspend until condition
Subrogate to "new Goal Name"
Determine .....(generic mental operator -- find from display, calculate, decide,etc.)

TRIGGERS
Message pattern templates based on blackboard contents

CONDITIONS include
context free CONTROL information:
if ..., repeat until ..., repeat n times, optional, etc.
domain-dependent EVALUATIVE information:
boolean statements based on blackboard message patterns

SELECTION RULES
Use Method...based on selection factors
Selection Rules:
if condition then Method 1...
if condition then Method 2 <with probability .x>

METHODS
Method 1 Name:
list of operators (and subgoals)

**Figure 2-3 COGNET Task Description Language**

Each subgoal or operator in the remainder of the COGNET task description may have a condition associated with it. Conditions must be expressed in terms that express context-free control constraints (such as if, when, until, etc.) or are completely evaluatable in terms of the blackboard contents. Thus, a condition could constrain an operator to be "applied 10 times", but not to be "repeated until a = b," unless "a=b" were a condition that could be evaluated strictly from the blackboard contents.

The COGNET task description language also provides a relatively fixed level of grain for the operator set. The most important observable operator is the 'PERFORM function' operator. COGNET focuses on human-computer systems, so the set of functions covered by the PERFORM construct is explicitly constrained to the set of commands or basic operations that are invokable from the workstation/interface being modeled. It should be noted that this is still a logical level, because it ignores the physical mode of implementation of the function. Thus, for example, when the operator

PERFORM display 'filename'

is encountered, it indicates that the person performs whatever interface function is needed to display the file 'filename' on the screen, whether this involves clicking on an icon, making a series of menu selections, or issuing a text-string command.

In addition to the problem-specific operators created by the PERFORM operator, there are six additional generic physical or observable operators. These are:

- **Point to [element/location on screen],** which refers to the action of using some pointing device to indicate a location or object/element on the workstation display;
- **Enter alphanumeric data** from a keyboard, keyset, or function key pad
- **Select displayed item** from the screen for further manipulation, and
- **Move pointing device** along specific path at a specific rate or set of rates.

These interface independent functions are similar to those used in the KLM.

In addition to the observable operators, the COGNET task description language provides for a fixed set of cognitive operators. The most important of these allow the common problem representation -- the blackboard -- to be modified as the person draws inferences, forms hypotheses, or manipulates information. These blackboard related operators are:

**POST**, which allows a specific piece of information or message to be placed on a specified blackboard panel and layer;

**UNPOST**, which allows a specific piece of information or message to be removed from a specified blackboard panel and layer; and

**TRANSFORM**, which allows an existing piece of information or message on the blackboard to be manipulated and replaced by a modified one.

Two additional cognitive operators provide the opportunistic attention flow capability. The SUSPEND operator allows a task to suspend itself, and create a local or temporary condition which will act as a trigger for the task to recapture control. This temporary condition is like all other COGNET goal/operator conditions, in that it must be expressed in a mix of domain independent control terms and blackboard-specific operations. The SUBROGATE operator allows the task to relinquish control and subrogate to another task. It defines the completion of the subrogated-to task as an implicit condition for recapturing of attention.

Finally, there is a more flexible generic cognitive operator which is called the DETERMINE operator. This allows for information manipulations that are associated with POSTs, UNPOSTs, and TRANSFORMs or that underlie observable actions to be explicitly stated and separated from the blackboard-manipulation operators. The DETERMINE operator is used to represent mental calculations, estimations, judgments, and other reasoning steps that are performed during the task based on the common problem representation.

These 10 operators are the only ones allowed in a COGNET task model, and provide a fixed yet domain-sensitive level of detail in the task description language.

Two additional features are enhanced from the original GOMS to support both individual differences and context-sensitive variations in task procedures. These are the METHOD and Selection Rule constructs. In the COGNET task description language, the METHOD construct is used somewhat differently from the original GOMS. METHODs in COGNET refer to rigid sequences of operators (and/or embedded subgoals), and are used in two ways. First, they can define domain-specific subtasks that are common to many COGNET network tasks. This allows these subtasks to be included in each task model only as the METHOD name. Second, they can define context-sensitive or individual variations in the procedure for accomplishing a given subgoal or step in a subgoal's procedure. In this case, multiple METHODS will be referenced in a given task model, along with SELECTION RULES for determining which METHOD is appropriate for a given individual/context. SELECTION RULES, as in GOMS, are if/then productions which define the conditions under which different methods should be selected. As in other aspects of COGNET, however, the conditions on which the rules are evaluated must be expressed in terms that can be evaluated form the blackboard contents. That is, the selection rules must represent patterns that can be found on the blackboard.

One final set of constructs used in COGNET task descriptions distinguish two orthogonal features of subgoals and operators in a task. The first feature is whether the subgoal/operator is required or optional for performance of the task. In a subgoal or operator sequence, some elements are critical while others can be ignored or eliminated with minimal impact on the overall problem solving process. In a COGNET task description, a given subgoal or operator is indicated as optional by placing it in parenthesis. When a subgoal is indicated as optional, all operators that comprise it automatically inherit this property, and therefore are not placed in parentheses. Instead, if an operator in an optional subgoal is indicated as optional, it is taken to indicate that it is optional with regard to the subgoal itself.

The second feature of operators and subgoals is that of order-dependence versus order-independence. In many tasks, some operations and/or subgoals must be pursued in a set sequence, while others can be done in a relatively arbitrary order.

COGNET assumes that all sequences are order dependent unless otherwise indicated. Subgoals or operators that are considered order independent (at their level of decomposition of the overall task goal) are indicated by being preceded by an asterisk in the task description.

### 2.3.4 Modeling perceptual processes

A problem with the task description language as defined above is that it links the global problem representation -- the blackboard -- strictly to cognitive operations performed within the individual tasks in the COGNET network. In terms of the Model Human Processor on which the GOMS is based, this leaves the COGNET with only cognitive and motor subsystems, since there are only cognitive operators (POST, UNPOST, TRANSFORM, DETERMINE, SUSPEND, SUBROGATE) and motor operators (PERFORM, POINT, SELECT, MOVE, ENTER). What is obviously missing is the operation of the perceptual subsystem, and any perceptual operators. In a human-computer interaction situation this is a critical failing, because much elementary information on the blackboard must be posted as the result of essentially perceptual events (e.g; observing a symbol appear or change on the display screen).

In the most general case, it would be desirable to include perceptual operators in the COGNET task description language that allow posting/unposting of information on the blackboard as the result of screen/workstation events. There is a problem in doing this, however, that arises from the very nature of the RTMT problem class. In RTMT domains, screen events appear as the result of unplanned, random, and/or hostile-agent actions. These unforeseen events are in fact the basis for the data-driven aspects of RTMT problem solving. The perceptual operator that perceives these events and posts them on the blackboard must therefore also be data driven. Unfortunately, such data-driven operators have no clear position in the goal-driven organization of the COGNET task language. A data-driven perceptual operator is equally applicable at all times and to all tasks in the network.

To allow for this type of data-driven perceptually-based blackboard access, a special type of construct was developed called the 'perceptual demon'. It consists of only a trigger and a POST operator, and is assumed to capture control and execute immediately whenever the triggering pattern or condition is observed. Unlike other task models, the conditions/triggers in a perceptual demon are not based on patterns on the blackboard, but instead are based on physical or workstation-based information, such as the registering of a datum on the display screen. The perceptual demons operate outside the flow of attention in the COGNET network -- which deals with the control of the cognitive processor -- because the control and sequencing of perceptual events is presumed to be separate and parallel in the human information processing architecture (A similar viewpoint is found in the MHP of Card et al., 1983). Thus, the perceptual demons form the link between physical events at the workstation and the registering of the information they contain in the workstation user's mental model of the problem.

# 3. MODELING METHODOLOGY

The COGNET model of ASW mission management was developed directly from experimental data on human performance in air ASW. Accordingly, data had to be obtained from expert TACCOs and analyzed in order to develop the model's content and structure. The subsections that follow discuss each of these two steps.

## 3.1 Data Collection and Experimental Method

The approach to modeling the TACCO's decision making environment was data-driven, building the model up from individual TACCO actions (operators) and the context in which they occur. Data were collected by having expert TACCOs perform a variety of realistic problems using an interactive ASW simulation environment and recording their performance.

### 3.1.1 Experimental Environment

To gather data on TACCO actions, an ASW mission simulation was implemented on a SUN 3/60 workstation (see Zachary & Zubritzky, 1988; Zachary, et al., 1988, for a complete description of the experimental environment). The simulation program models the independent actions and movements of the entities involved in an ASW mission, such as the target and aircraft movements, and sensor behavior. It operates in real-time and graphically represents the movements of the entities on an emulated TACCO workstation.

The heart of the experimental environment is an interactive simulation of a general but stylized Air ASW mission. The simulation is general enough to allow a wide variety of experimental problems to be solved. However, in order to constrain the complexity of the experiments and the data collection, some details of the ASW problem were simplified or removed. First, the behavior of the acoustic sensors has been smoothed, and their error generalized. Second, the behavior of the non-acoustic sensors has been approximated by simple 'cookie-cutter' models. Third, the key real-world problems of buoy drift and plot stabilization have been removed by modeling buoy drop as perfectly true and vertical and eliminating any buoy drift. Other computations regarding the compensation for navigational error and buoy locational error have similarly been eliminated in the name of simplicity. Fourth, no channel management is required. What remains are the purely tactical functions (i.e., load and drop buoys, integrate contact information, direct the aircraft, and plan attack and weapon drops).

The interface to the mission simulation is an emulated TACCO workstation (shown in Figure 3-1) which resembles the actual TACCO crewstation as much as possible, both in form and in function. This allowed the participants to make maximum use of their existing expertise in ASW, mission management, and minimized the amount of learning required by experimental participants.

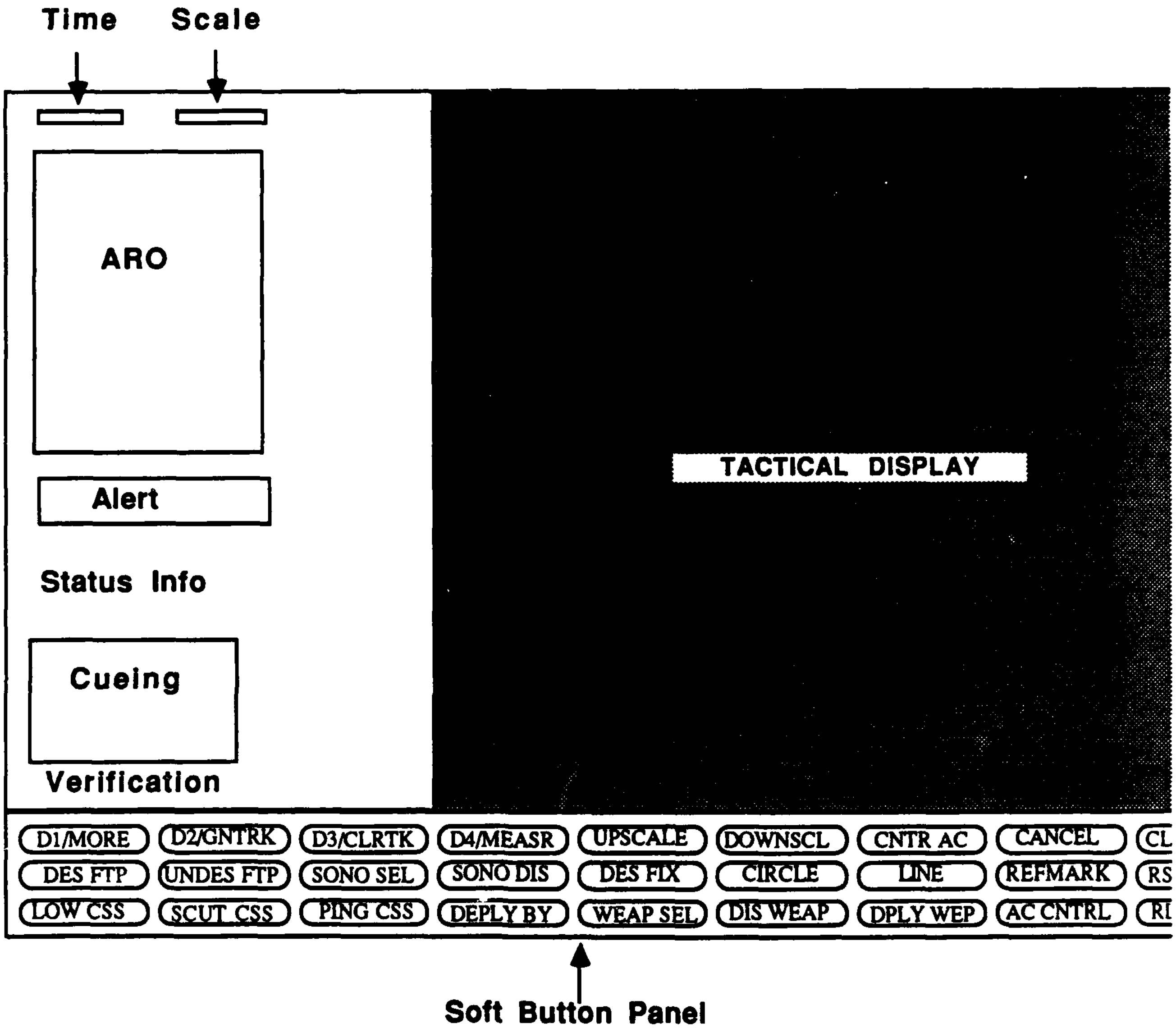


**Figure 3-1. Simulation Interface**

The simulation has the capability of using mission scenarios which differ on the basis of mission objectives, target behavior, environmental conditions of the ocean, and sensor capabilities. Experimental problems were designed to vary all of these parameters in order to observe their effect on TACCO strategies. Each problem consisted of one to three hours of activity. A set of five partial mission simulations have been defined by an expert TACCO. These simulations focus on the initial stages of a mission in which the TACCO set out an initial or search pattern of sensors and monitors it waiting for an initial contact. These partial simulations are pre-defined and replayed (faster than real-time) at the beginning of an experimental session to establish a context for the participant. At the point of first contact, the replay is discontinued and control is turned over to the user, and data collection begins.

The experimental environment includes a data capturing/analysis component that allows for the collection of three distinct but interrelated types of data: 1) TACCO actions, 2) TACCO intentions while performing sets of actions, and 3) Situational context of TACCO actions. Each is discussed in further detail below.

In terms of the simulation, all TACCO actions can be classified as button presses, key presses, and mouse inputs on the graphical screen. Each time a TACCO takes an action, it is recorded (in binary format) with a time-stamp and all parameters relevant to the particular action. For example, if the TACCO deploys a buoy (a button press action), this action will be recorded along with the buoy's position on the screen, the type of buoy, and the functional status of the buoy. This binary file of user action data is then transformed (via the TIMELINE program) into a narrative timeline for ease of use in building the GOMS models. Figure 3-2 shows an example of a timeline file.

To gather data on TACCO intentions, post-experiment protocol data are gathered with the use of the replay capability of the simulation. With this capability, the simulation can use the user action file as input in order to replay the entire simulation with the TACCO responses included. The mission is replayed with the TACCO and the experimenter present, and the TACCO is asked to describe his/her thoughts and intentions in performing a particular action or group of actions. This protocol data proved to be invaluable in delineating tasks from the stream of user actions provided by the timeline.

In addition to the user action file, the simulation also builds a symbol data file during the course of the mission which holds records of all the tactical symbols on the screen and the states they are in (e.g., their current position, time created, etc.) at the occurrence of every TACCO action or any change in the tactical situation. This file provides information about the current situation so that the experimenter can assess the context in which each TACCO action takes place. An example from a symbol data file is shown in Figure 3-3.

| | |
|---|---|
| 01:14:27.0 | Enters Undesignated Expendable FTP at 21.30, 25.00 |
| 01:14:32.8 | Changes MPD to scale 128 with center at 22.20, 27.80 |
| 01:15:00.2 | Deletes FTP at location 21.30, 25.00 |
| 01:15:17.0 | Enters Undesignated Expendable FTP at 24.20, 52.20 |
| 01:15:39.0 | Enters Undesignated Expendable FTP at 27.30, 81.70 |
| 01:15:55.0 | Draws circle with center at 20.30, 21.70 and radius of 29.00 (via hook point plus radius) |
| 01:16:20.0 | Changes MPD to scale 128 with center at 23.20, 29.30 |
| 01:16:24.4 | Changes MPD to scale 128 with center at 2.70, 32.70 |
| 01:16:59.7 | Draws circle with center at 20.70, 21.80 and radius of 58.00 (via hook point plus radius) |
| 01:17:44.7 | Changes aircraft altitude to 2000.00 feet |
| 01:18:10.7 | Enters Undesignated Expendable FTP at 19.20, 51.20 |
| 01:18:16.1 | Deletes reference circle |
| 01:18:23.0 | Deletes FTP at location 27.30, 81.70 |
| 01:18:27.5 | Deletes FTP at location 24.20, 52.20 |
| 01:18:37.2 | Draws circle with center at 20.70, 21.30 and radius of 29.00 (via hook point plus radius) |
| 01:18:51.0 | Disarms all armed buoys |
| 01:19:59.0 | Enters Undesignated Expendable FTP at 25.30, 50.20 |
| 01:20:08.3 | Deletes FTP at location 19.20, 51.20 |
| 01:23:27.0 | Changes MPD to scale 128 with center at 3.00, 48.50 |
| 01:24:29.1 | Draws circle with center at 22.70, 49.80 and radius of 2.00 (via hook point plus radius) |
| 01:24:43.8 | Deletes FTP at location 25.30, 50.20 |
| 01:24:52.6 | Enters Undesignated Expendable FTP at 22.70, 50.20 |
| 01:25:20.1 | Enters Undesignated Expendable FTP at 22.50, 52.80 |
| 01:37:07.2 | Arms DIFAR for Short/Deep |
| 01:37:20.6 | Arms DIFAR for Short/Deep |
| 01:37:33.7 | Enters Undesignated Expendable FTP at 22.20, 79.20 |
| 01:38:44.6 | Changes MPD to scale 64 with center at 3.00, 48.50 |
| 01:39:01.3 | Changes MPD to scale 64 with center at 24.10, 51.10 |

This listing contains a time-stamped record of user actions. Each action is described in textual form referencing the tactical display coordinates of the action, if applicable.

**Figure 3-2 Example Timeline**

```
STIME 1:14:45: 0 ARCRFT -41.17 -8.72 BEAR 61.65 ALT 5000.00 SPEED 216.00 ONSCREEN
*****
STIME 1:15: 0: 0 ARCRFT -40.43 -8.32 BEAR 61.65 ALT 5000.00 SPEED 216.00 ONSCREEN
*****
STIME 1:15: 0: 0 CENTER 22.17 27.83 NAUTICAL MILE SPAN 128

     SEQ# X Y
BUOY   1124 -28.67  1.92 DIFAR
BUOY   1108 -8.75  2.17 DIFAR
BUOY   1092 11.42  2.33 DIFAR
BUOY   1076 31.33  2.17 DIFAR
BUOY   1060 50.92  2.42 DIFAR
BUOY   1044 40.83 21.92 DIFAR
BUOY   1028 20.83 21.75 DIFAR CONTACT ON  6.10 BEARING AND  0.00 RANGE
BUOY   1012  0.75 22.00 DIFAR
BUOY    996 -19.75 21.92 DIFAR
BUOY    980 -39.08 22.17 DIFAR
CIRCLE  292 -19.75 22.67 OF RADIUS  0.17
CIRCLE  212 -39.33 22.50 OF RADIUS  0.33
TARGET      20.49 52.47 BEAR -134.89 DEPTH 350.00 SPEED  7.45
ARCRFT     -40.43 -8.32 BEAR 61.65  ALT 5000.00 SPEED 216.00
*****
```

This listing is divided into segments, separated by asterisks. Each segment represents a single 'snapshot' of the context. The first item of the segment indicates the simulation time to which it refers. Each time a user action is taken or a symbol has changed on the screen, the complete contents of the tactical display are recorded. In addition, even if no other changes occur, a record is made of the aircraft position every 15 seconds. This example shows two instances of aircraft position updates and one of the tactical display content recording.

**Figure 3-3 Example Symbol file snapshot**

### 3.1.2 Participants

Five expert TACCOs participated in the data collection. The minimum criterion for expertise was more than one deployment of operational experience as TACCO on a P-3C, since it meant that the individual had completed TACCO training and had a period of intensive application of TACCO skills. The participants included three TACCOs with recent operational experience and two who had at least two years since their last operational mission. They had commanded missions in a variety of geographical areas, and on several different versions of the P-3C aircraft, and thus posed a good sample of the P-3C TACCO community.

### 3.1.3 Procedure

Participants were asked to solve two or more ASW mission management problems to completion. The first problem was presented as a sample problem, to allow the participants to become familiar with the simulated workstation. Following the sample problem, the participant would solve one or more additional problems, as time allowed. All participants completed at least two problems, with two participants completing three problems, one completing four problems, and one completing five problems. This resulted in five sample trials and twelve experimental trials.

The experimental problems represented a range of realistic operational scenarios that made varying demands on the TACCO. Three main factors which are reported by domain experts to affect decision making--target motion, environmental conditions, and resource availability-- were varied.

Target motion pattern is one variable that affects the difficulty of the mission. Three typical target behaviors were modeled in the simulation and used in the problems, as follows:

1) transiting --a target that is moving on a general heading as if transiting from one location to another location. Within this general behavior, the target makes periodic changes in heading, speed, and depth to make detection and tracking by ASW systems more difficult.
2) holding -- a target that is attempting to patrol on-station in a given piece of ocean for an extended period of time. The target has no general long-term heading, and its behavior involves many moves in random segments, even to the extent of crossing and criss-crossing its own track. In its movement segments, it uses variation in speed, heading and depth to evade detection.
3) sprint and drift -- a target that is moving in a general direction, but which is using extreme variations in speed to evade detection. This type of target may move very rapidly on one movement segment, and then move very slowly for some period thereafter. These 'drift' periods reduce acoustical emissions and make detection and sustained tracking very difficult.

Variations in environmental conditions also present different demands to the TACCO. Environmental conditions were modeled by creating acoustical propagation loss profiles that presented the TACCO with:

- a direct path contact only;
- a direct path contact and one convergence zone; and
- a direct path contact and two convergence zones.

The first condition, with direct path contact only, represents a situation in which contact with a target is difficult to gain, but relatively easy to prosecute once gained. It also requires (typically) expenditure of a large number of sensors. The second condition, with one convergence zone, provides a situation in which initial contact is easy to gain,

but much effort must be devoted to disambiguating the contact and localizing it. The third condition, which is increasingly rare (due to the new generation of quieter targets) makes initial contact easier still to achieve, but presents much greater difficulty in localization and disambiguation of the contact. It also stresses time resources, because it implies a larger search area and requires more buoys. By far the most common environment is direct path with one convergence zone; therefore the majority of our problems used that condition.

Two dimensions of resource availability have been defined, time and expendable sensors, which depend on the initial sonobuoy search patterns (which vary widely in the time and sensors required) and time when initial contact occurs. A high time-resource availability was created by using an initial search pattern that is rapidly deployed (specifically, the line) and/or by allowing initial contact to occur on other patterns before the full pattern was deployed. A low time-resource availability was created by using patterns that required much time (5-6-5, cross, containment) and allowing all or nearly all the pattern to be deployed before the initial contact was gained.

| Problem number | Target Motion | Resources Available Time | Resources Available Sensors | Environment | Search Pattern |
|---|---|---|---|---|---|
| 1 (sample) | Sprint & Drift | high | low | One CZ | Cross |
| 2 | Holding | low | low | One CZ | 5-6-5 |
| 3 | Transiting | high | high | One CZ | Line |
| 4 | Sprint & Drift | high | low | One CZ | Containment |
| 5 | Transiting | high | low | No CZ | Wedge |

**Table 3-1 Experimental Problem Characteristics**

The characteristics of the five problems used are shown in Table 3-1. The sample (i.e., first) problem was designed to be a problem of medium difficulty. The second and fourth problems were of moderate to high difficulty, while the third and fifth problems were of low to moderate difficulty. Thus, the problems varied both in difficulty and in the type of decision-making requirements imposed. Each problem had a package of briefing material that is similar in content, although much simplified, to the briefing material a TACCO normally received prior to an ASW mission.

An experimental session began by familiarizing the participant with the experimental environment. This involved reading a written description of the simulation and emulated TACCO workstation, followed by questions and answers. The participant then had a chance to "play with" the simulation interface in performing the sample task. Once the familiarization and practice was completed, the individual performed one or two experimental problems. Individuals who had time returned and completed additional problems in a second session.

Each problem involved three steps. First, the participants read the briefing package describing the mission and saw the replay of the initial context up to the point of first contact. Then, the participant completed the mission performing TACCO functions by interacting with the mission simulation through an emulated TACCO

station, much as an operator would do on an actual Air ASW mission. The data set recorded during each problem by the experimental environment contained an average of 400 actions and 40,000 display items for each experimental trial. Following completion of the experimental trial (i.e., simulated mission), the executed problem was replayed with the participant, who was asked to think aloud, describing intentions and motives while performing actions or action sequences, and expectations about the consequences of those actions. The experimenter interacted with the participant in this replay, stopping execution and directing specific questions to the participant's thought processes. Thus the process was an example of a 'question answering' verbal protocol, as created by Graesser and Murray (1989), and resulted in an average of one-hour of question answering protocol data per experimental trial. This verbal protocol data was tape recorded for use in conjunction with the analysis of the keystroke-level data. Because the specific questions asked by the experimenter are also, in essence, part of the data analysis process, the details of the question answering protocol methods are discussed in greater detail below.

## 3.2 Data Analysis and Model Construction

The models of TACCO mission management were built from all of the data sources listed above, in a multi-stage analysis process. This process consisted of a task decomposition stage, a task definition stage, and a problem representation stage. The latter two of these were conducted iteratively and in parallel.

### 3.2.1 Decomposition

The initial stage of the analysis decomposed the problem domain into a set of tasks that formed the COGNET task network. The primary data for this analysis was the recorded data on the keystroke-level interactions between the TACCO/participant and the experimental environment in a given experimental trial. The verbal question-answering protocol provided both a secondary source of data and a first stage of analysis. The protocol data is considered secondary because it is derived from a replay of the real-time problem solving process. However, because the verbal protocol involved analytical questions posed by the experimenter to the subject, it also represented a first stage of analysis of the primary data. A review of the differences between question-answering and thinking aloud protocol can help further clarify this dual role of the protocol generation process.

In the classical 'thinking aloud' protocol method, the experimenter allows the participant free reign in recounting the thought processes, and reconstructs the lexicon, semantics, and problem solving processes from after-the-fact analysis of the transcribed protocol (e.g., as described in Waterman and Newell, 1971; Ericsson and Simon, 1984). However, in computer-human interaction domains, the role of the dynamics of computer displays on cognitive processes may not be made explicit in such an unstructured protocol. Key features of a cognitive process may remain undiscussed, as the participant focuses on the internal thought process and not on the perceived environment (i.e., the computer and its behavior). Graesser and Murray (1989) argue this point strongly, and make a case for an experimenter directed

protocol, one in which the experimenter focuses the introspection on features of interest in the modeling processes. Clancy (1987) makes a similar argument in a more general knowledge engineering context. In this research, preliminary pilot trials with pure thinking aloud vs question answering protocols clearly favored the latter. Each experimental trial was therefore followed with an intensive question answering protocol.

Given that the ASW mission management is an expert task (and not an intuitive one, such as cryptarithmetic or text editing), domain knowledge was assumed in advance to play an important role in the computer-interaction strategies. Accordingly, the experimenters all acquired a thorough familiarity with the existing vocabulary of the domain, plus some knowledge of goals, and actions in the domain by reviewing training, and doctrinal literature. Thus, because the experimenter already understood the lexicon and semantics of the task, the verbal data collection approach could concentrate on a question-answering protocol method, with clearer and more focused experimental goals. Specifically, in the protocol following each trial, the experimenter sought to identify which task the participant was performing at each point in time (with the level of grain supplied by the subject or by default), and why the participant was performing that task at that time. Earlier pilot trials had proved that an unstructured thinking-aloud protocol alone was insufficient to achieve these goals.

The pilot trials also pointed out the difficulties of recording data about discussions of time-dependent problem solving computer-interactive decision processes, and led to the creation of the automatic time-line generation feature of the experimental environment. The question-answering protocol involved constant references to specific events in the mission evolution, screen/display conditions, and to sequences of both. Synchronizing elicited protocol data with these real-time and context markers proved an insurmountable data recording problem without some larger framework for data recording. It was to solve these problems that the timeline and symbol snapshot files were created. A timeline of the human-computer interactions could act as a background for recording data from the question-answering protocol, and for making precise time-dependent linkages and connections between verbal data and display events.

The sequence for the generation, recording, and analysis of verbal data thus proceeded as follows. First, the experimenter would have a participant solve a specific mission simulation. Upon completion of the simulation, the experimenter would convert the user action file into a timeline and print a hardcopy of the timeline. This conversion/printing process required less than one minute, about the time required to name and store the experimental trial data files. Using the timeline essentially as a notepad, the experimenter would then replay the mission to the participant, and request the participant to articulate the thoughts and intentions that motivated the actions as they were originally taken. The experimenter would, when necessary, stop the replay and question the participant directly as to his goals, task choices, etc.

The initial phase of the data analysis therefore resulted in a complex annotation of the timeline to indicate the segments of activity associated with specific tasks. There are times where, on replay, participant could not reconstruct which specific task he was performing (in which case alternative reconstructions were maintained), or had no idea of what he was doing. ( This was labeled as undefined. ) Figure 3-4 shows an

example of this process. Figure 3-4a shows a basic unannotated timeline, and Figure 3-4b shows the kinds of annotations made during the question-answering protocol. In this case, the participant indicated that in the beginning of this five minute sequence, he was engaged in setting up and executing a tactic designed to gain a possible MAD contact. At approximately minute 36, he indicated that he had formed a hypothesis that the target had just made a closest point of approach to a sonobuoy, and was marking this hypothesis with a screen annotation . After completing this task with the entry of the Designate Fix symbol at time 37:09, he indicated that he was beginning a tactic intended to narrow the uncertainty in the target's actual location relative to the fix symbol.

A related analysis, undertaken during the Question-Answering protocol procedure, was identifying the context associated with each task shift. In some cases, the information was spontaneously generated by the participant. Continuing with the example from Figure 3-4, the participant noted that at his decision to begin planning a MAD run over the target was based on a change in the display. Specifically, he noted that two bearing lines that had previously diverged had now proceeded to intersect at a location where a target could feasibly be located. These context events were also annotated on the timelines. When participant volunteered no basis for task switching , it was solicited via a directed question. The shift from the MAD run task to the target hypothesis formation task, for example was associated with several events on the screen, plus the completion of the MAD planning task. When questioned, the participant in this case indicated that the hypothesis was formed at this point because he has observed a sudden, radical shift of a bearing line on a sonobuoy. Similarly, once the target fix was entered, the participant responded that the completion of that task had allowed him to consider other possible tasks, and he had decided to begin the narrowing of uncertainty task because of additional signal information that had been received from one of the sonobuoys in contact with the target. The elicitation of these context cues resulted in an annotated timeline as shown in Figure 3-5.

After all experimental trials were completed, the final definition of the tasks for the COGNET model was undertaken. The specific tasks identified in each participants timeline protocol were then compiled, and the task lists were compiled across subjects and correlated for commonality. This was done because some tasks are idiosyncratic, or applicable only to specific theater of operations, while other are commonly used and/or doctrinal. General or commonly applied tasks often had specific names and were considered standard procedure for the situation at hand. For instance, performing a MAD run is a standard TACCO task undertaken when the TACCO has a hypothesis of target location. This task is implemented by lowering the aircraft and entering two fly-to-points (FTP's) in the area where the TACCO hypothesizes the target to be. (As shown in Figure 3-4a, all of these actions are part of the group of actions

Figure 3-4a. Initial Timeline

| | |
|---|---|
| 00:34:02 | Enters Undesignated Normal FTP at 31.70, 17.90 |
| 00:34:10 | Designate target fix at 32.10, 18.20 |
| 00:34:23 | Changes MPD to scale 16 with center at 32.10, 16.20 |
| 00:34:27 | Changes MPD to scale 8 with center at 32.10, 16.20 |
| 00:35:01 | Enters Undesignated Normal FTP at 32.80, 18.40 |
| 00:35:19 | Changes aircraft altitude to 300.00 feet |
| | |
| 00:36:00 | Designate target fix at 32.10, 17.90 |
| 00:36:17 | Enters Undesignated Normal FTP at 31.60, 17.70 |
| 00:36:35 | Arms DIFAR for Short/Deep |
| 00:37:09 | Designate target fix at 31.90, 17.80 |
| | |
| 00:38:03 | Draws circle with center at 32.30,17.50 and radius of 0.80 |
| 00:38:21 | Enters Undesignated Normal FTP at 31.30, 17.10 |
| 00:38:38 | Arms DIFAR for Short/Deep |
| 00:39:34 | Enters Undesignated Expendable FTP at 32.00, 16.80 |
| 00:39:43 | Deletes FTP at location 31.30, 17.10 |

Figure 3-4b. Timeline With Annotations

| | | |
|---|---|---|
| 00:34:02 | Enters Undesignated Normal FTP at 31.70, 17.90 | **Protocol: MAD run over target (Standard Tactic)** |
| 00:34:10 | Designate target fix at 32.10, 18.20 | |
| 00:34:23 | Changes MPD to scale 16 with center at 32.10, 16.20 | |
| 00:34:27 | Changes MPD to scale 8 with center at 32.10, 16.20 | |
| 00:35:01 | Enters Undesignated Normal FTP at 32.80, 18.40 | |
| 00:35:19 | Changes aircraft altitude to 300.00 feet | |
| 00:36:00 | Designate target fix at 32.10, 17.90 | **Protocol: Hypothesis that target is at CPA** |
| 00:36:17 | Enters Undesignated Normal FTP at 31.60, 17.70 | |
| 00:36:35 | Arms DIFAR for Short/Deep | |
| 00:37:09 | Designate target fix at 31.90, 17.80 | |
| 00:38:03 | Draws circle with center at 32.30,17.50 and radius of 0.80 | **Protocol: Narrow area of probability (Nonstandard tactic)** |
| 00:38:21 | Enters Undesignated Normal FTP at 31.30, 17.10 | |
| 00:38:38 | Arms DIFAR for Short/Deep | |
| 00:39:34 | Enters Undesignated Expendable FTP at 32.00, 16.80 | |
| 00:39:43 | Deletes FTP at location 31.30, 17.10 | |

**Figure 3-4. Timeline with Protocol Annotations**

| | |
|---|---|
| **DISPLAY: CONTACT BEARING INTERSECTION** | **Protocol: MAD run over target (Standard Tactic)** |
| 00:34:02 Enters Undesignated Normal FTP at 31.70, 17.90 | |
| 00:34:10 Designate target fix at 32.10, 18.20 | |
| 00:34:23 Changes MPD to scale 16 with center at 32.10, 16.20 | |
| 00:34:27 Changes MPD to scale 8 with center at 32.10, 16.20 | |
| 00:35:01 Enters Undesignated Normal FTP at 32.80, 18.40 | |
| 00:35:19 Changes aircraft altitude to 300.00 feet | |
| **DISPLAY: RADICAL BEARING SHIFT** | **Protocol: Hypothesis that target is at CPA** |
| 00:36:00 Designate target fix at 32.10, 17.90 | |
| 00:36:17 Enters Undesignated Normal FTP at 31.60, 17.70 | |
| 00:36:35 Arms DIFAR for Short/Deep | |
| 00:37:09 Designate target fix at 31.90, 17.80 | |
| **DISPLAY: ADDITIONAL BEARING INFORMATION** | **Protocol: Narrow area of probability (Nonstandard tactic)** |
| 00:38:03 Draws circle with center at 32.30,17.50 and radius of 0.80 | |
| 00:38:21 Enters Undesignated Normal FTP at 31.30, 17.10 | |
| 00:38:38 Arms DIFAR for Short/Deep | |
| 00:39:34 Enters Undesignated Expendable FTP at 32.00, 16.80 | |
| 00:39:43 Deletes FTP at location 31.30, 17.10 | |

**Figure 3-5 Timeline With Protocol and Display Context Annotations**

defined by this and other participants as a MAD run). Other tasks were identified by the TACCO as being self-devised but effective for the situation. An example of a nonstandard tactic is the Narrow Area of Probability task shown at the end of Figure 3-5. One particular TACCO used this strategy to make successive refinements of target location, and it was very effective; however, none of the other participants ever used it and it was not part of doctrine. The task was therefore ultimately discarded as idiosyncratic. Tactics that were completely idiosyncratic were not included in the final list. However, most individual variations were of the form of modifications or elaborations of standard or more widely performed tasks. In these case, these variations were treated as 'a kind of' the more general task.

After this was done for all timelines, the final list of (common or shared) tasks was subjected to process of abstraction and generalization. Particularly, tasks were grouped into similar areas and assessed on dimensions of

"is task A a kind of task B (or vice versa)"

"is task A a part of task B (or vice versa)"

"are tasks A and B both instances of some more abstract task 'c' ?"

As result of this process, four tasks were defined, along with fourteen more detailed tasks, each of which was a part of one of the more generalized tasks. These tasks are listed and discused further in Section 4 and Table 4-1 below. As a final check, the task decomposition was reviewed with domain experts, and found acceptable and

complete. The fourteen low-level tasks were defined as the basic tasks in the COGNET representation of Air ASW Mission Management, because of the more detailed representation they provided.

The modeling of these tasks in the COGNET task description language was then undertaken, although as shown below, this process was highly iterative with the modeling of the global problem blackboard structure.

### 3.2.2 Detailed Modeling Analysis

The next two steps in the analysis were detailed modeling of the individual tasks identified from the decomposition analysis, and development of the global blackboard representation that integrated problem solving activity across the tasks. These two steps were conducted essentially in parallel, and in an iterative manner, with each iteration increasing the level of detail and conformity within and between the task models and blackboard representation. The general structure of this iterative process is given below:

1)The first pass through this iterative process involved developing an initial goal decomposition of the various tasks in the COGNET network, and a set of levels of abstraction in the blackboard structure.

2)With these in place, a second iteration of analysis increased the level of detail in the goal decomposition in the individual task models to the point that a (provisionary) complete goal structure was identified. In parallel, a complete (but also provisionary) set of panels and panel levels was established for the blackboard structure.

3)In the third iteration, the observable operators were embedded into the goal structures to produce GOMS-like descriptions of the tasks. Where there were identifiable individual or group differences in methods for achieving some of the goals, these were separated and modeled as separate METHOD constructs, with SELECTION RULEs identified for choosing among them. In parallel, the blackboard message semantics that was associated with the observable processes in the various tasks were introduced into the blackboard structure.

4)Next, the COGNET cognitive operators were introduced into the GOMS-like task descriptions, along with cognitive conditions on the pursuance of the various goals in the task models. These cognitive operators and conditions were expressed in the semantics already defined for the blackboard. However, attempts to do so frequently pointed out the need for further clarification or modification of the semantics and message structure of the blackboard. This analysis step continued until there was a consistent problem representation structure defined in the individual task model cognitive operators and goal conditions and in the blackboard representation.

5)A final step involved linking the blackboard and task models in two additional ways, via the perceptual demons that posted initial information on the blackboards, and via the triggers, suspensions, and subrogations relations among tasks.

The main data used in these analyses were the annotated timelines developed in the decomposition analysis, supplemented by occasional reviews of the (audio-taped) question-answering protocol data. The timelines were separated into segments indicating the task instances identified earlier (i.e., as marked in Figure 3-4 and 3-5). In cases where a task was interrupted by another task, and later revisited and completed, the two (or more) timeline segments were collected and integrated, with the intermediate task(s) removed. Then, all of these segments were linked with the appropriate sequence of display/symbol snapshots (as shown in Figure 3-3), and labeled as being instances of one the fourteen COGNET tasks. In addition, each task segment was labeled with the subject number, and information on the scenario characteristics used to generate it (e.g., '1CZ environment, holding target, high source noise levels').

Portions of timelines that could not be allocated to any task, or to one task with more than 50% certainty were discarded. Although no formal statistic was calculated, only a small portion of the overall timeline set was discarded in this fashion. These task-instance keystroke level sequences were then analyzed with the iterative procedure described above to develop the individual task and blackboard components of the COGNET model. A partial example of this process is discussed below.

### 3.2.3 Some Detailed Examples

The iterative process of detailed model development can be seen in the development of the model for the task identified as "Investigate Convergence Zones." This task occurs in virtually all missions where the environmental conditions permit convergence zone contact. The task is undertaken when the TACCO obtains a sensor contact that has little or not corroborative data (including contact history). Such contacts are likely to be CZ contacts, and thus require a set of sensors to be deployed to investigate the Convergence Zone. The first step, as indicated above, began by collecting all annotated examples of this task, and subgrouping them according to subject and problem conditions. The audio tapes of the sections of the question-answering protocols corresponding to these task segments were also reviewed. From the examination of the timeline annotations and audio tapes, a simple high-level goal structure emerged. This structure consisted of three high level and sequential goals within the task, as indicated below:

Goal: Display CZ on Screen

Goal: Mark Line-of-Interest Intersection with CZ on Screen

Goal: Build Pattern in Plotted CZ

The first goal, "Display CZ on Screen", referred to the need of the TACCO to locate the area to be investigated. CZs are not automatically displayed, and must be manually drawn by the TACCO. In addition, the TACCO must determine the specific sonobuoy whose CZ is to be investigated, normally, the one having current or recent contact. The CZ area is itself a large anulus covering hundreds of square miles, the

TACCO must identify the portion of the zone to be investigated, and display that area on the display screen. This flows into the second goal, "Mark Line-of-Interest Intersection with CZ on Screen." This reflects the further need of the TACCO to physically mark or otherwise denote on the display screen the exact area of the CZ that is to be investigated. Then, once the precise area to be investigated, the last goal is define a pattern of sensors that will, when deployed, achieve the desired investigation of the area.

Several elements of the blackboard structure emerged from this initial level of task modeling. The concept of an "area of interest" or AOI was used by all subjects to refer to a logically-defined area of ocean that was of interest because it could contain the source of a sensor contact. In this case, the AOI type is a CZ AOI, meaning that an approximate intersection of some line (either a bearing line or a visual centroid of several bearing lines) and a CZ around a specific sonobuoy. There were other kinds of AOIs, however, and they were frequently discussed by subjects in recounting their strategies. Thus, a separate blackboard level was defined to contain AOI information. Another related structure was the contact itself. There are many kinds of sensors and each may have several kinds of contacts, but TACCOs uniformly derived AOIs from contacts. Thus, a separate contact level was also assigned to the blackboard.

In the second iteration of the analysis, the goal structure was elaborated by more detailed analysis of the various example timelines of this task activity (and their associated protocol data). Several aspects of detail structure emerged. It was discovered that the task really has two parallel halves, one to investigate each possible convergence (assuming a 2CZ environment). This leads to two higher level goals, *"investigate outer CZ" (if there is one), and "investigate inner CZ,"* which were pursued sequentially (in the order given). Additionally, one added subgoal emerged in the detailed analysis, the goal of focusing the display on the CZ AOI. This identified the need to insure that the scale and center of the display screen gave the TACCO a clear view of the area in which the pattern was to be planned out. The resulting complete goal structure for this task was finalized as given below:

GOAL Investigate CZs

- GOAL:Investigate Outer CZ
  - Goal: Display outer CZ on Screen
  - Goal: Mark LOI Intersection with CZ on Screen
  - Goal Focus Display on CZ AOI
  - Goal: Build Pattern in outer CZ
- GOAL:Investigate Inner CZ
  - Goal: Display Inner CZ on Screen
  - Goal: Mark LOI Intersection with CZ on Screen
  - Goal Focus Display on CZ AOI
  - Goal: Build Pattern in inner CZ

The third step in detailing this task model was to embed the observable operators. This was done by further annotating each timeline segment to identify the specific

operations that were part of each subgoal. In most cases, this was easy to do with the timeline, annotation, and protocol data. Where confusion or ambiguity arose, the goal structure was reviewed, and in many cases, modified. For purposes of this example, only one portion of the model is presented to this level of detail, the "Mark LOI Intersection with CZ on Screen" subgoal. In examining the actual computer interactions needed to achieve this goal, two distinct methods were observed. In one method, the line-of-interest already intersected with the CZ as drawn on the screen, and the TACCO simply marks the point of intersection (as a hedge against unexpected loss of contact). This is termed the Mark POI Method. In the other method, the line of interest does not intersect the CZ, or must be constructed as an 'average' of several bearing lines. This is termed the Mark LOI Methods. Modeling these sequences with the COGNET modeling language resulted in the following sequence:

GOAL: Mark LOI Intersection with CZ on screen...*optional, more likely if LOI definition is not = contact bearing or if LOI does not intersect CZ midline*
*use* ***Marking Methods****...based on individual variation*
Methods:
**Mark LOI**
Hook contact buoy or average point of bearing lines near contact buoys
Perform DRAW LINE
Hook end of bearing line or average point of ends of contact bearings
**Mark POI**
Hook intersection of LOI and CZ midline
Perform REFMARK

In the fourth step in modeling this task, the cognitive operators (as defined in Section 2) were embedded into the model. The "Build Pattern in inner CZ" goal is used as the example case for this step. This goal, at the end of the third step, had several alternative Methods, representing the different approaches that could be used to lay out a sensor pattern. The model at this point was then:

GOAL: Build Pattern in Inner CZ
use **Layout CZ Pattern Method**... *based on individual variation and mission factors: buoy resources (BR), confidence in existence of CZ in question (C), time remaining on station (TOS)*
Selection Rules:
If C high, BR high, and TOS either high or low; then use **Method A** (probability .5) or **Method B** (probability .5)
If C high and BR and TOS low; then use **Method C**
If C low, TOS high and BR high; then use **Method C**
If C low, TOS high, but BR low; then use **Method D**
Methods:
**Method A: Standard Line**
**Method B: Standard Line with Wings**
**Method C: Reduced 2-buoy**
**Method D: Reduced 1-buoy**

The cognitive operators, in this case, focused on the TACCOs integration of his intention to create a sensor patter into his problem representation. That is, the TACCO started pursuing this subgoal by mentally noting that a pattern was intended; if

interrupted, he used this declaration as an anchor for later returning to and completing the goal. Thus, the first cognitive operation in this sequence was a POST operation:

> <u>Post</u> "1st CZI Pattern around buoy 'contact buoy number' planned" on patterns level of Situation BB

Similarly, as the TACCO completed the sequence of actions needed to map out the pattern and cue it for automatic deployment, the intention to create a pattern changed to a belief that the pattern was laid out and prepared for deployment. Thus, the sequence of observable operators was followed by a TRANSFORM operator of the form:

> <u>Transform</u> "1st CZI pattern around Buoy 'contact buoy number' planned" to "1st CZI Pattern around buoy 'contact buoy number' mapped" on patterns level of situation BB

This operation also had the effect of integrating the pattern with the rest of the representation of the problem situation, by relating the pattern to an existing sensor from whose contact data the new pattern is based. The type and content of these POST/TRANSFORM operators also contributed to the refinement of the message structure defined on the blackboard panels.

The fifth and final step in detailing the model of this task was the definition of the task trigger and associated task relationships. This task was found to have a complex trigger. Two obvious conditions for this task to capture attention were;

1) an environment in which there was perceived to be one or two CZs, and
2) the existence of an current passive sensor contact.

These conditions were necessary, but on further examination, were not sufficient. If this were, this task would be initiated each time there was a new sensor contact in a CZ environment. This is clearly not the case. A more detailed examination of the context of the observed occurrences of this task showed that it was undertaken only when there was no clear hypothesis as to the target's location. This reflect the fact that main purpose of this task is to obtain a direct path contact; if one was clearly already present, the task would be unnecessary. Thus, an added aspect of the task trigger was a "no Direct Path hypothesis" condition. The trigger for this task was thus defined as:

***(CZ1 or CZ2 AOI on target BB) AND (no DP hypothesis on target BB)***

The task also had one interaction with another task, the Plot Acoustic Environment Features task. As part of the first subgoal -- Locate Outer(Inner) CZ on Display -- the TACCO needs to view the physical location of the CZ of interest on the screen. In some cases, this feature will already be drawn (as a result of other tasks). If not, however, the TACCO will suspend this task and initiate the Plot Acoustic Environment Features task. This relation is indicated by the following subrogation operator:

> <u>Subrogate</u> to"Plot Acoustic Environmental Features"...*if CZ not already plotted*

At this point, the task model was complete (as contained in Appendix A). This process was simultaneously pursued for all tasks in the COGNET model. Additional features of the final resulting model are discussed in the following section.

# 4. ASW MISSION MANAGEMENT MODEL

This section presents the COGNET model of Naval Air ASW mission management that was built from the methods and data described in the preceding section. "Mission management" is a term that was coined to encapsulate the TACCO's role in the later portions of an Air ASW mission. These are the portions which involve a protracted period of real-time multi-tasking on the part of the TACCO. To further bound the scope of the model presented below, it is useful to review the initial stages of an Air ASW mission (which are not explicitly included in the our model) and to contrast them with the latter phases which are.

The role of the Air ASW TACCO in an operational mission really begins at the pre-flight briefing well prior to aircraft take-off. In this briefing, and subsequent pre-flight planning periods, the TACCO receives key information about the mission, and forms expectations and global plans for how to prosecute the mission. This planning continues into the en-route portion of the mission, which, in the case of a patrol aircraft like the P-3, may take several hours. When arriving on-station, the TACCO proceeds to deploy environmental sensors from which an updated profile of the oceanographic and acoustical conditions is developed. Following this, a preplanned pattern of passive acoustic sensors is mapped out (and possibly adjusted according to the updated oceanographic data), and deployed. The TACCO begins to build mental model (or in COGNET terms, to populate a blackboard representation) of the current mission throughout the preflight, enroute and search pattern deployment periods. The mission management does not consider these phases, however, because they proceed in a non-real time or non-time-critical fashion. In addition, they involve little or no multi-tasking, because the TACCO can not yet receive any data on the target submarine. After the initial deployment of the search pattern, however, this is no longer the case. Once the first sensor contact is gained, then the TACCO must begin a protracted prosecution of the contact that is in part goal-driven (based on training, experience, and standard doctrine) and in part data-driven (based on the specific sensor data received). The overall goal of this prosecution is to form a highly accurate hypothesis as to the location, depth, course and speed of the target submarine. "Highly accurate" is operationally defined as sufficiently accurate to launch against the target with a standard (torpedo) weapon. In peacetime, of course, the TACCO does not place an attack but merely attempts to maintain this level of knowledge and track the submarine, ideally 'handing off' the target track to a relief platform for continued tracking. Mission management, as used in this model, covers the period of the mission between the initial sensor contact and either total loss of the target or development of a highly accurate hypothesis regarding target location, depth, course, and speed. This period, which normally ranges from 30 minutes to two or more hours, is the most RTMT portion of the overall mission.

In the following sections, the task structure of the mission model is introduced first. The problem representation blackboard is then presented, followed by a summary of the major perceptual demons that link the blackboard with the outside world. Next, there is detailed consideration of the individual task models and their internal problem-solving strategies, followed with a discussion of the ways in which attention flows between and among the Air ASW tasks.

## 4.1 Tasks Comprising Air ASW Mission Management

The analysis of the experimental problem data identified two levels of decomposition of mission management as defined above. The first is a relatively coarse breakdown of TACCO problem solving into four aspects of the mission which the TACCO manages. The four primary aspects or partial mission management goals are:

- maintaining a complete picture of the tactical situation,
- managing control of the aircraft,
- hypothesizing/inferring the activities of the target, and
- managing the patterns of sonobuoys deployed.

In the second level of decomposition, each of these general areas is broken into specific tasks which are performed as part of each aspect of problem solving. A total of fourteen of these lower level tasks were identified (see Table 4-1). The lowest level fourteen tasks were selected for use in the COGNET model, because they represented more focused and well-defined units of activity. Attention is shared among these four aspects generally, and specifically between and among the fourteen local goals and the specific tasks into which the four areas are decomposed. Each high-level aspect is discussed in detail below, along with the specific goals/tasks comprising it.

| | |
|---|---|
| Control Aircraft | Control Sensor Suite |
| *Position in Area of Interest* | Manage Sonobuoy Resources |
| Preposition for Expected Events | Broaden Initial Contact |
| Maneuver for MAD | Investigate Convergence Zones |
| | Expand Pattern for Contact Continuity |
| | Deploy Sensor/Pattern |
| Maintain Situational Awareness | Hypothesize Target Activity |
| Plot Acoustic Environmental Features | Identify Area of Interest |
| Review Overall Situation | Develop Target Fix |
| | Gain Attack Criteria |
| | Determine Target Track |

**Table 4-1 Task Areas and Individual Tasks in ASW Mission Management Model**

Maintaining the complete picture of the tactical situation involves two lower level tasks. One concerns compensating for the limitations of the computer-generated view of the tactical situation. The tactical screen can display a 'window' on the overall situation in geometric scales of 2 (i.e., a 2nm view, a 4nm view, an 8nm view, etc.). Most activities following initial contact require a view that crops much of the current pattern of sensors, thus requiring the TACCO to constantly pan over the larger situation or (more commonly) to upscale and downscale to gain the bigger picture.

Another task within this goal is that of inferring and projecting (via drawing functions) the effect of the oceanographic sound propagation conditions onto the

computer display of the tactical situation. The need for this arises from a physical phenomenon called the convergence zone or CZ phenomenon. The existence of CZs makes the use of passive sonobuoys more difficult that would initially appear. An acoustic sonobuoy can 'hear' sound directly propagated from an emitting source over a small distance; this is called its direct path (DP) detection range (e.g., 2-5 nautical miles). Because of ducting of sound underwater, there may be a small annular region quite distant from the DP zone in which detection may also occur. This is called a CZ. There may be no, one, or possibly two CZs in any given acoustical environment. The presence of CZs creates a complex pattern of potential detection regions in a field of sonobuoys (see Figure 4-1). Directional passive sonobuoys also provide a bearing to the target, but this bearing contains error, and does not help disambiguate whether the sound originates from the DP zone or from some CZ. Thus, the TACCO must often plot manually the specific location of various DP and CZ regions associated with sensors having contact.

Managing control of the aircraft is a context-sensitive activity, in that it is performed to different criteria as the mission prosecution evolves. This goal is decomposed into three lower-level tasks. The first position the aircraft for some immediate tactical action (e.g., deploy a sensor). The second requires the TACCO to position the aircraft for possible future action (e.g., remain behind the target so that a weapon can be deployed on the target track when attack criteria are gained) or so as to be prepared to take advantage of some expected opportunity (e.g., remain near a sonobuoy on which contact is expected). The TACCO must constantly compensate for the time it can take for the aircraft to move to the point where it can undertake a desired tactical action. In many cases, a piece of tactical information is temporally transitory, and can be capitalized on only if the aircraft can be maneuvered into a proper position in some time window. These time windows are often on the same scale as the aircraft motions, so there is constantly a real possibility that an opportunity will be lost because of aircraft maneuvering. There is also a reverse special case, in which the TACCO must be careful to avoid flying the aircraft directly over the target at low altitude so as not to alert the target of the aircraft's presence. The third task involves maneuvering the aircraft to be in optimal position for gaining MAD contact. The gaining of MAD contacts, while possible at anytime, is enhanced by performance of specific aircraft maneuvers that position the aircraft in an area where MAD contact is more likely. Thus maneuvering to gain MAD contact has both tactical and aircraft control significance.

128 Square Mile Tactical Plot

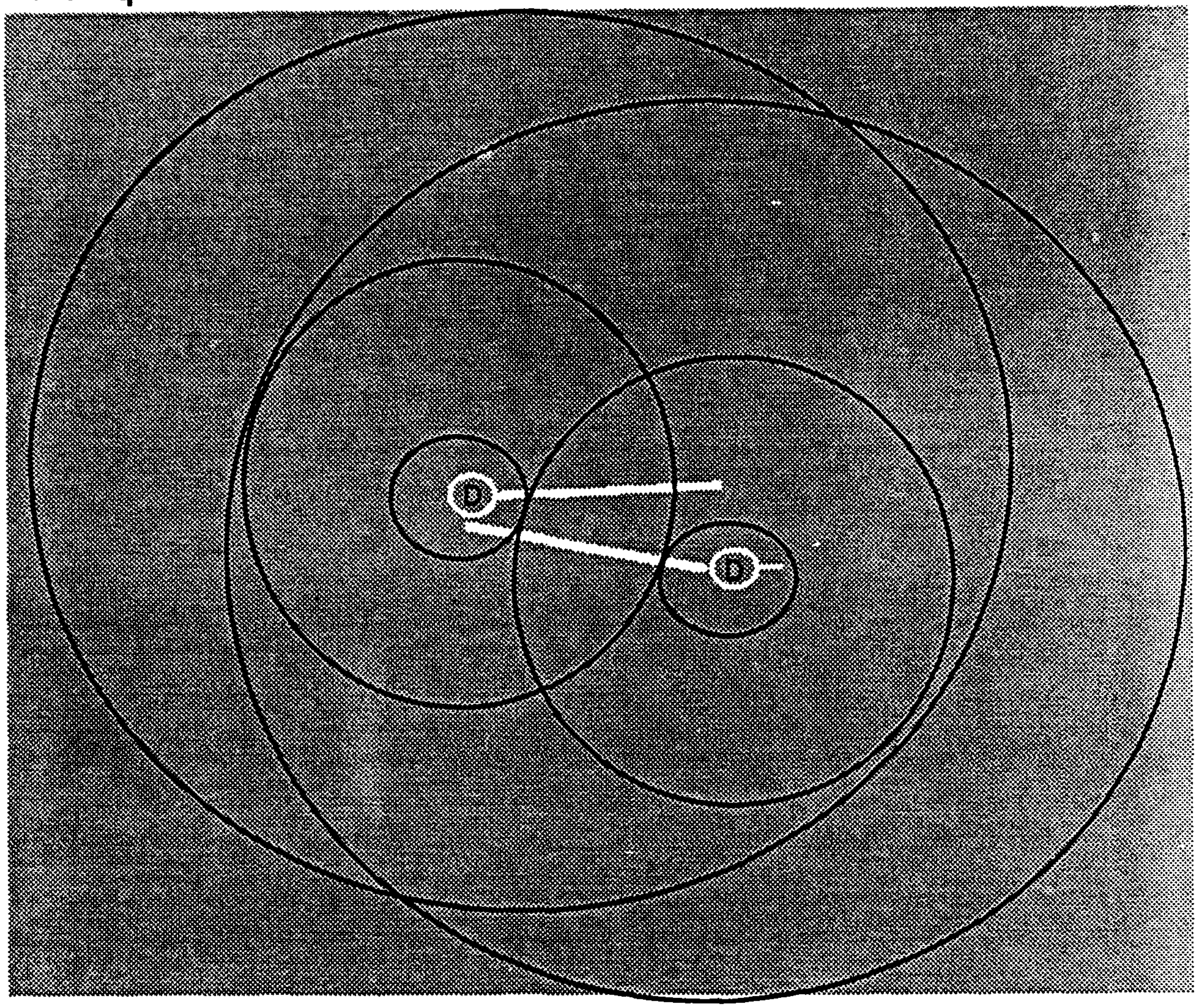

**Figure 4-1. Convergence Zones and Intersections for Two Sonobuoys**

Once an initial sensor contact is made with a potential target submarine, the TACCO repeatedly will hypothesize about target behavior. In this, the operator performs tasks that lead to the formation and testing of many hypotheses about the target. The four lower-level tasks within this general aspect involve hypotheses incorporate varying degrees of detail. The most general hypotheses simply deal with 'sensible' regions associated with a sensor or sensors having a contact. These regions are termed 'areas of interest.' Areas of interest (or AOIs) are not locational hypotheses per se because a given contact of pattern of contact can generate multiple AOIs. TACCOs do not treat AOIs as locational hypotheses but rather as building blocks for constructing locational hypotheses. Locational hypotheses or target fixes are developed through a series of sensor, aircraft, and/or inferential process, depending on the problem evolution and context. A target fix is noted explicitly on the display screen by some symbol, usually the TACCO-entered 'target fix' symbol. More

refined hypotheses about target behavior include course and speed as well as location. These hypotheses, called target tracks, are again developed in problem-specific ways. Both target fixes and target tracks are treated as inherently uncertain hypotheses; frequently TACCOs maintain multiples of each at any given point in time. A low- or no-uncertainty track hypothesis is required to achieve the overall goal (weapon placement). The development of this type of hypothesis is termed attainment or gaining of attack criteria, and represents the most constrained target hypothesis. It too may be performed in a variety of ways depending on problem situation and evolution.

The management of the sensor suite is also undertaken in a context sensitive fashion, and involves five different tasks. Several refer to the selection and deployment of new partial patterns of acoustic sonobuoys. When a key sensor contact is gained, a set of sensors will be deployed around it to broaden the period of time in which that contact will be held (by the original or one of the surrounding sensors). Another task concerns the planning of a pattern of sonobuoys in an AOI that is associated with a CZ around a sensor currently having contact. A combination of these first two tasks concerns the expansions of a pattern of sonobuoys in a direction associated with a target track hypothesis, to maintain continuity of contact. Another goal of this 'contact continuity' task is maintaining the integrity of the pattern already in place. Sensors deployed early in the mission and later holding good contact may have to be replaced as their life span expires. Similarly, sensors that were placed for a given purpose must sometimes be replaced by others at a slightly different location to compensate for target movement. Throughout all these sensor planning tasks, the TACCO must manage a finite set of sonobuoy resources and adapt the plans accordingly.

The final task in this area of mission management concerns the actual deployment of sonobuoy patterns. Planned sensor locations are entered by the TACCO via graphical symbols in the tactical screen. The symbols, call Expendable Fly-To Points, serve two purposes. They tell the pilot to fly the aircraft to that location, and they tell the aircraft machinery to drop or deploy a sensor when that geographical location is achieved. However, the TACCO must still go through a separate sets of actions to tell the aircraft which type of sensor must be deployed, as well as its particular settings. Because all sensors in a pattern typically are of a common type and setting, the TACCO must intersperse instructions to select and configure sensors between the capturing of Expendable Fly-to-Points and the deployment of a sensor at each. This task is referred to collectively as deploying a Sensor or Sensor Pattern.

## 4.2 Problem Representation

As described in Section 2 above, the integrative problem representation is formalized as a blackboard structure. A general blackboard structure may contain separate partitions, or panels, to deal with information about different aspects of the problem. Furthermore, each panel is decomposed into a hierarchy of levels containing one or more hypotheses or partial solutions constructed by the individual tasks. The ASW mission management blackboard structure consists of two panels-- the target panel and the situation panel -- each representing separate yet highly interrelated aspects of the problem solution space. The target panel contains

information about the TACCO's evolving hypotheses about target behavior; while the situation panel contains an understanding of the evolving prosecution. These two solution aspects are constructed relatively separately, but draw on each other as sources of data.

### 4.2.1 Target Panel

The target panel is divided into six levels of abstraction which are used by the TACCO to process sensor data about a possible target. Each level represents increasingly more refined hypotheses about target behavior.

The lowest level is the contact level. It contains hypotheses and/or perceptual events denoting sensor contact. When a contact 'appears' as a display symbol on the TACCOs tactical display, a perceptual demon is immediately triggered which posts the information on the contact level

The next level of abstraction involves the application of knowledge about the sensor which produced the contact, properties of the specific acoustic environment, and the overall situation to define areas of interest that may arise from sensor contacts and/or other (more abstract) information on the blackboard.

The third hierarchical level on the blackboard represents a special kind of area of interest, a direct path region around an acoustic sensor, MAD, or radar contact. Direct path information represents a maximally crude locational hypotheses about a target.

The fourth level refers to more precise locational hypotheses that are based on various combinations of other hypotheses, knowledge, and/or sensor data from other levels on the blackboard. Usually, it is necessary to fuse information from multiple contacts to obtain a location hypothesis.

The fifth level of the target blackboard refers to directional hypotheses about the target. These may be mixed levels of abstraction, from general directional information gleaned from prior knowledge to specific directional hypotheses inferred from sensor data on the blackboard.

Finally, the sixth and highest level on the target blackboard refers to fused directional and locational hypotheses, which are referred to as tracks. These hypotheses often correspond to moving track symbols generated by the TACCO via the workstation software. It is not uncommon for a TACCO to have five or more of these track hypotheses for a single target.

Figure 4-2 shows the blackboard target panel contents gleaned from a specific experimental trial. The arrows show the general flow by which information is posted and transformed, indicating the mixed directions in which information is processed on the blackboard. Initially, there is only a weak directional hypothesis of an expected southwesterly motion of the target. This hypothesis would have been developed prior to take off, as the result of intelligence information in the pre-flight briefing. Because the pre-contact mission phases are not included in the present model, such hypotheses are dealt with as assumptions. That is, the model assumes that relevant information about the mission that would have been developed prior to the beginning of the mission management phase is already posted on the blackboard at the start of this phase. (This assumption is generally more important to the situation blackboard than to the target blackboard).

Returning to Figure 4-2, after deployment of the initial search pattern, one sensor (that on channel 19) from that pattern gains a directional contact. That contact is posted on the contact level of this blackboard panel as "New DIFAR on 19 at t1, bearing 30" ,indicating a new directional contact was first obtained on a DIFAR sensor using channel 19 at time t1, with the directional bearing at 30° to the sensor having the contact. The TACCO incorporates this information into a representation through a perceptual event, i.e., by perceiving the contact and associated bearing line as they appear on the tactical screen. Thus, this initial posting in the model is done via a perceptual demon (see 4.5 below).

The posting of a new contact on the blackboard triggers the Identify Areas of Interest task, which determines the possible areas of interest associated with the new contact and posts them as AOIs on the AOI level of the target panel. Only one of these is shown in Figure 4-2, the Convergence Zone Area of Interest that is associated with the sensor on Channel 19. The overall pattern of information on the blackboard at this time triggers two additional tasks, the Broaden Initial Contact task, which deploys a sonobuoy pattern around the sensor on channel 19, and the Investigate Convergence Zone task, which deploys a pattern of sensors in the Channel 19 CZ AOI. The layout of this pattern is influenced by the existing motion hypothesis already on the direction layer of the target panel. Because of this hypothesis, the Investigate Convergence Zone task maps out the pattern slightly to the southwest of where it would have been laid out (in a no-information case). One of these sonobuoys in the CZ pattern is deployed using Channel 22, and soon gains a contact at time t2. This new contact is also posted on the blackboard by a perceptual demon. The influence of the existing directional hypothesis on this contact is indicated by the arrow from the directional hypothesis to the contact message.

The new contact on Channel 22 again triggers the Identify AOI task, which this time implies that only a direct path contact is reasonable for the sensor using Channel 22. This information, in combination with the continuing CZ AOI on sensor 19, further leads to a direct path hypothesis being posted on the Direct Path layer of the target panel. After a short time, the TACCO further follows this by making an initial locational hypothesis about the target at the location of the intersecting bearing lines for sensors on channels 19 and 22. This locational hypothesis is denoted as (w,z) and time t2 on the blackboard.

The new pattern of information on the blackboard at this time triggers the Maneuver for MAD task, through which the TACCO attempts to develop a more precise locational hypothesis via a combined DIFAR and MAD contact. Through this task, the aircraft does obtain a MAD contact, at location (x,y) and time t2, as indicated on the contact layer. This contact is then combined with the directional information and direct path hypothesis from the sensor on Channel 22, to yield a hypothesis that the target was near x,y at time t3. Moreover, this locational hypothesis is further combined with the previous directional hypothesis and the previous locational hypothesis (i.e., w,z at t2) to generate a refined directional hypothesis, i.e., that of movement along bearing 200°. This is also followed by creation of a moving track symbol on the screen, anchored at location x,y at time t2, and moving on course 200°.

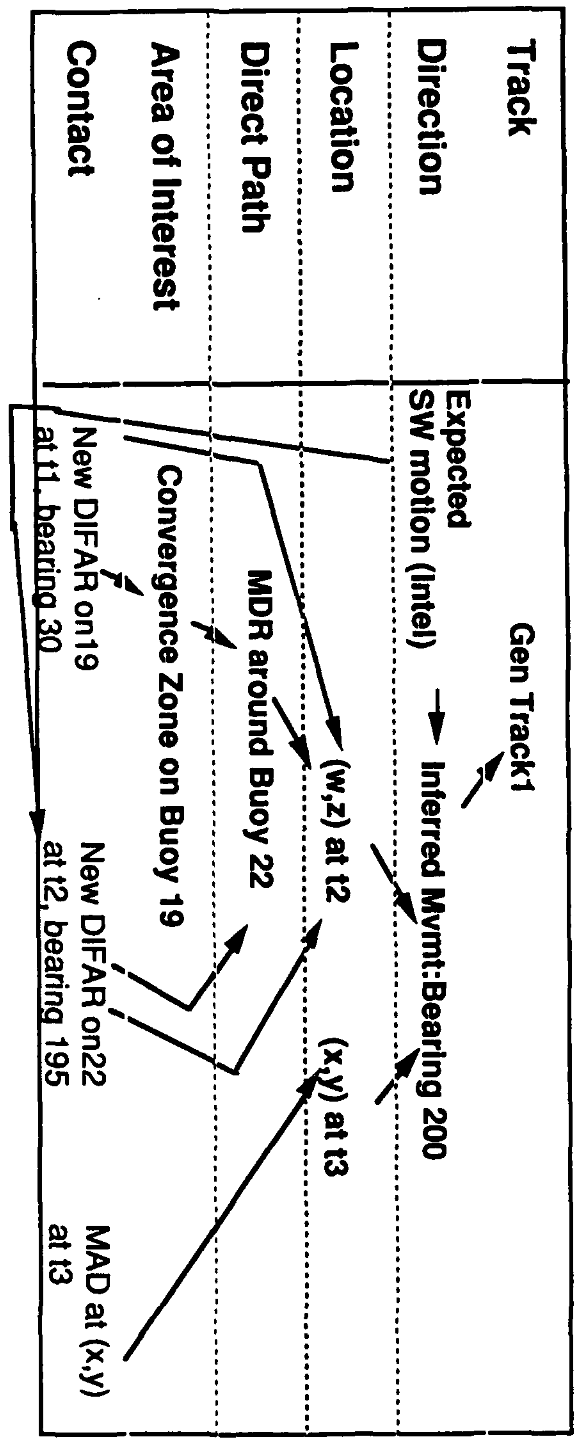


**Figure 4-2 Target Panel of Blackboard showing Example Information Dynamics**

### 4.2.2 Situation Panel

The Situation panel of the blackboard contains information about the individual elements (e.g., the aircraft and sonobuoys) and features (e.g., environmental properties) of the tactical situation. It also is divided into six levels as shown in Figure 4-3. In this figure, as in Figure 4-2, the contents show data from a specific experimental trial.

The first two levels, off-screen elements and tactical display, contain situational information that is represented on the TACCO workstation. Most of these are data that are plotted on the tactical screen, such as where the aircraft is and where all tactical display symbols are located. These display symbols include buoy locations, fly-to-points, contact symbology (e.g., bearing lines, MAD contact symbols), 'chalkboard' information such as reference circles, reference lines, and reference marks; and other symbology.

These two levels provide a means for representing the spatial relationships among elements over the complete tactical area. The tactical display level contains those elements which are visible on the TACCO's tactical display screen, and hence, are of most immediate interest. The off-screen elements level contains all other elements over the whole mission area. Because the tactical display has a 'zoom/pan' organization, there is often symbology that is not currently on the screen. These data, represented as they were last viewed by the TACCO, are contained on the off-screen elements layer. In addition to the actual tactical display contents, the tactical display layer indicates other items of information that are perceived from the workstation, such as the center point of the display and its scale, the aircraft bearing and speed, and other status information.

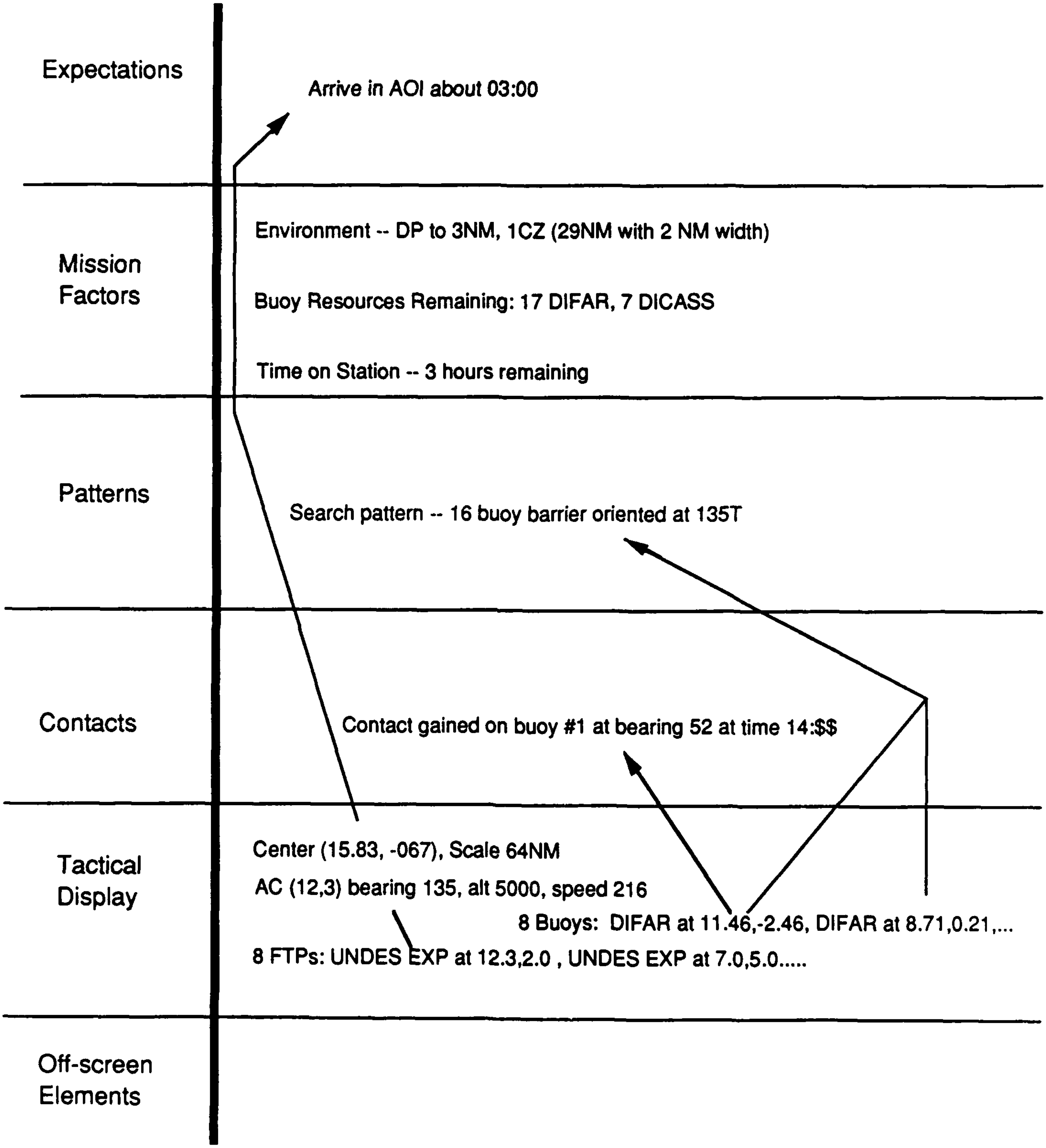


**Figure 4-3. Organization of the Situation Blackboard Panel Showing Contents During A Specific Operator Trial**

The third and fourth levels are contacts and patterns. The contact level contains a time-stamped history of the sensors that have gained and/or lost contact. If at some point the TACCO gets information conflicting with a current target hypothesis, that individual may refer to the contact history to reevaluate what is known about the target. The pattern level contains a record of all buoy patterns planned or in use. Different patterns are used for different mission phases and are selected based on the TACCO's current target hypothesis and various mission factors. An important aspect of the pattern level is that patterns are usually defined relative to other aspects of the situation. For example, a convergence zone investigative pattern reflects a standard geometry that is adjusted to reflect features of the acoustical environment and laid out relative to another sensor that has the contact being investigated. Thus, an entry on the pattern level will indicate pattern type (e.g., investigative, contact-broadening) and also status (e.g., planned, partially-in-water, fully-deployed, partially dead, dead). however, it will additionally contain links to other sensors or features on the tactical display layer (e.g., sensor locations) and the mission factors layer (e.g., environmental features).

The fifth level contains data/hypotheses about mission factors, including environmental information about how sound will be propagated in the mission area, resources remaining, etc. The TACCO's current hypothesis about the environment influences how that person interprets contacts. Normally, these hypotheses are developed prior to the mission management period at either the pre-flight briefing or the initial on-station phase where environmental sensors are deployed and analyzed. However, patterns of sensor contacts may conflict with posted environmental information, which may in some cases cause the TACCO to revise environmental hypotheses. Resources remaining influence what strategies will be used for target prosecution. For example, the number of buoys remaining influences the selection of buoy patterns to use.

The sixth level contains expectations about future events and when they are likely to occur. For example, when the TACCO enters a Fly-to-Point (FTP), a steering command to the pilot (or in this case, the simulation), he creates an expectation about when that FTP will be captured, which influences his decisions about what he can accomplish until that time. Such expectations can lead to suspensions of currently active tasks, as well as triggers for suspended tasks to recapture control.

Figure 4-3 shows the contents of the situation blackboard at the beginning of the mission management period of a specific experimental trial. The TACCO has begun to deploy the search pattern, which is indicated as a barrier type pattern containing sixteen sonobuoys with center at location (a,b) and orientation of 135°. This information is 'inherited' from the preceding mission phases, as is much of the information posted at the Mission Factors level. Specifically, the expectations about the sound propagation environment -- with a 3nm direct path range and a single convergence zone centered at 29nm with a 2nm radius -- were all established at the pre-flight briefing and confirmed by initial on-station environmental sensors. The buoy resources remaining -- 17 DIFARs and 7 DICASS -- are updated by perceptual demons each time the TACCO views the resources table on the ARO portion of the workstation.

The contents of the tactical display indicate that the initial search pattern is half deployed, with 8 buoy symbols listed and 8 Expendable Fly-to-points also listed. Each

time a symbol changes on the visible portion of the screen, this level of the situation panel is updated by an appropriate perceptual demon. Since this is early in the mission and the screen is at a relatively high scale (64nm), there are no off-screen symbols at the lowest level.

The TACCO is engaging in the Deploy Pattern/Sensor task, and is deploying the initial search pattern. The aircraft has just captured and deployed a sonobuoy on Channel 8 (the eighth buoy in the search pattern) and is now proceeding to the next FTP at location (12.3, 2.0). This has resulted in the TACCO generating an expectation that the next aircraft will arrive at the next FTP in about 3 minutes. However, the contacts level of the situation panel also indicates that the buoy on Channel 1 has gained contact with a potential target. This event, which will also be posted on the Target panel of the blackboard, will eventually trigger the Broaden Contact and Investigate Convergence Zone tasks.

## 4.3 Perceptual Demons

Some information is posted on the blackboard as the result of essentially perceptual events, e.g; observing a contact bearing come up on the screen. To account for this type of information access, a special type of GOMS model was developed called the perceptual demon. It consists of only a trigger (as defined in Section 2) and a POST operator, and is assumed to capture control and execute immediately whenever the triggering pattern or condition is observed. In the ASW Mission Management model, the triggers in the perceptual demons are linked with display events in the ASW Mission Management experimental environment (Zachary & Zubritzky, 1988). Thus, all display events in the experimental environment can be used as triggers for the perceptual demons. The current list of the perceptual demons associated with the Mission Management model is shown in Table 4-2.

Buoy gains contact ==>

if DIFAR contact

POST : "New DIFAR Contact at time [mission-time] on Buoy [Channel No] with Bearing [bearing] and Strength [high/low] on Target/Contacts and Situation/Contacts"

if LOFAR contact

POST: "New LOFAR Contact at time [mission-time] on Buoy [Channel No] and Strength [high/low] on Target/Contacts and Situation/Contacts"

if DICASS contact

POST: "New DICASS Contact at time [mission time] on Buoy [Channel No] indicating range of [range in yards] and Bearing [bearing] on Target/Contacts and Situation/Contacts"

if CASS contact

POST: "New CASS Contact at time [mission time] on Buoy [Channel No] indicating range of [range in yards] on Target/Contacts and Situation/Contacts"

if MAD contact

POST "MAD Contact at time [mission time] at [x,y] on Target/Contacts and Situation/Contacts"

if RADAR contact

POST: " RADAR contact at time [mission time] at [x,y] on Target/Contacts and Situation/Contacts"

Buoy loses contact ==>

if DIFAR contact

POST: "DIFAR Contact on Buoy [Channel No] lost contact at time [mission-time] on Target/Contacts and Situation/Contacts"

if LOFAR contact

POST: "LOFAR Contact on Buoy [Channel No] lost contact at time [mission-time] on Target/Contacts and Situation/Contacts"

Bearing shift of DIFAR contact ==>

POST: "DIFAR Contact on Buoy [Channel No] bearing shift at time [mission time] to bearing [bearing] on Target/Contacts and Situation/Contacts"

Change in signal strength ==>

if DIFAR contact

POST: "DIFAR Contact on Buoy [Channel No] changed strength to [high/low] at time [mission time] on Target/Contacts and Situation/Contacts"

**Table 4-2. Perceptual Demons**

if LOFAR contact

POST: "LOFAR Contact on Buoy [Channel No] changed strength to [high/low] at time [mission time] on Target/Contacts and Situation/Contacts"

Change in aircraft location ==>

POST: "AC (x,y) bearing [bearing] alt [feet] speed [TAS] on Situation/Tactical Display

Capture of FTP ==>

if no expenditure

UNPOST: "[type] at [x,y]

if expenditure

TRANSFORM: "[type] at [x,y]" TO "[symbol type] at location [x,y] at time [time] on Situation/Tactical Display"

Symbol moved off-screen by recenter or downscale ==>

TRANSFORM: [type] at [x,y] on situation/tactical display" to "[type] at [x,y] at time [time] on Situation/Off-screen Elements"

Expenditure of buoy resources ==>

if DIFAR

TRANSFORM: "Buoy resources remaining: [number] DIFAR" to "Buoy resources remaining: [number -1] DIFAR"

if DICASS

TRANSFORM: "Buoy resources remaining: [number] DICASS" to "Buoy resources remaining: [number -1] DICASS"

**Table 4-2. Perceptual Demons (contd)**

## 4.4 Individual Task Models

Models were built of all fourteen tasks indicated in Table 4-1. Each model was constructed within the COGNET task description language introduced into Section 2. Emphasis was put on those tasks that are not concerned with attack criteria per se to avoid potential classification difficulties. Each model contains a mixture of cognitive and behavioral (i.e., human-computer interaction) operators, but not always in the same proportion. Table 4-3 shows an extreme case, the initial portion of the model for Identifying an Area of Interest. This model is triggered essentially any time there is a new sensor contact posted on the target panel of the blackboard, and reflects a totally cognitive process in which contact data are transformed into areas of interest for further examination. As can be noted from Table 4-3, once this task is triggered by a new contact message, it produces no externally observable actions. Instead, this task model captures an inferential process by which other blackboard information is applied to the contact datum to generate specific areas of interest. For example, it the TACCO was told at the pre-flight briefing (or so concluded while on-station) that there was one convergence zone, a CZ area of interest will be POSTed at the AOI level of the target blackboard. This model demonstrates how a COGNET task differs from a conventional blackboard knowledge source (as defined, for example, by Nii, 1986). While the canonical knowledge source performs a single transformation of a blackboard, this particular task can make several changes and transformations onto the blackboard. This model also demonstrates how information on multiple blackboard panels is used together in a single task.

Table 4-4 shows a portion of a task that involves more directly observable human-computer interaction, the Broaden Initial Contact task. This table shows the portions of the task that focus the display on the area in which the pattern is to be plotted and that draws the actual pattern on the screen as a series of fly-to-points. This sequence is relatively inflexible, and so differs very little from instance to instance and from TACCO to TACCO. It represents a segment of human-computer interaction that can is readily observable and recognizable from this type of model of the task.

The Appendix provides a complete listing of the models of these two tasks, plus models of :

Position in Area of Interest,

Plot Acoustic Environmental Features

Review Overall Situation

Manage Sonobuoy Resources

Deploy Sensor/Pattern

plus complete part-task models for ten multi-task Methods that are used by these Tasks. These methods are:

Sonobuoy Select

Weapon Select

Designated Fly-to-Point

Undesignated Fly-to-Point

Designate Fix

Aircraft Control
Generate Track
Clear Track
Draw Circle
Draw Line

**GOAL: IDENTIFY AREA OF INTEREST...*any new [sensor type] contact posted on contact level of target BB***

GOAL: Identify Acoustic AOI...*if new DIFAR, LOFAR, CASS, or DICASS contact*

GOAL: Identify passive AOI...*if new DIFAR or LOFAR contact*

Determine [contact buoy number] from contact level of target BB

Post "DP AOI on 'contact buoy number' on AOI level of target BB

Post "BTmBn AOI on 'contact buoy number' on AOI level of target BB...*if Bottom Bounce on mission factors level of situation BB"*

Post "CZ1 AOI on 'contact buoy number' on target BB...*if 2CZ or 1CZ on mission factors level of situation BB*

Post "CZ2 AOI on 'contact buoy number' on target BB...*if 2CZ on mission factors level of situation BB*

Transform "New [buoy type] contact at time [mission-time] on Buoy [Channel No] with Bearing [bearing]" into "[buoy type] contact at time [mission-time] on Buoy [Channel No] with Bearing [bearing]"on contact level of target BB

GOAL: Identify active AOI...*if CASS or DICASS contact*

Determine [contact buoy number] from contact level of target BB

Post "CASS AOI with MDR [distance] on buoy [Channel No]" on AOI level of target BB...*if CASS contact*

Post "DICASS AOI with bearing [bearing] and with MDR [distance] on buoy [Channel No]" on AOI level of target BB...*if DICASS contact*

Transform "New [buoy type] contact at time [mission-time] on Buoy [Channel No] with Bearing [bearing]" into "[buoy type] contact at time [mission-time] on Buoy [Channel No] with Bearing [bearing]"on contact level of target BB

**Table 4-3. Portion of Identify Area of Interest Task Model**

GOAL: Focus tactical display on AOI

<u>Perform</u> UPSCAL...*until AOI visible on display*

<u>Perform</u> RECENTER (contact buoy)...*if contact buoy not near center of screen*

<u>Perform</u> DOWNSCAL...*until AOI fills display*

GOAL: Build prudential pattern

<u>Subrogate</u> to "Plot Acoustic Environmental Features"...*if DP not already plotted*

GOAL: Enter FTPs

<u>Hook</u> 1st point on DP circle

<u>Perform</u> UNDES FTP (expendable, hooked location)

<u>Hook</u> 2nd point on DP circle

<u>Perform</u> UNDES FTP (expendable, hooked location)

<u>Hook</u> 3rd point on DP circle

<u>Perform</u> UNDES FTP (expendable, hooked location)

<u>Hook</u> 4th point on DP circle

<u>Perform</u> UNDES FTP (expendable, hooked location)

**Table 4-4. Portion of Broaden Initial Contact Task Model**

## 4.5 Attention Flows Among Tasks

As described in Section 2.3.2 above, the flow of attention between and among these tasks is complex, and reflects a combination of both forward directed and backward directed control. To some degree the TACCO's attention process is opportunistic, for example, as that individual takes advantage of slack periods to perform routine tasks or seizes upon aspects of the tactical situation to prepare resources for expected busy periods. In terms of observed behavior, there are three different ways in which control flows between tasks during a complete problem instance:

- capturing of control,
- suspension of control,
- subrogation of control.

In general, capturing of control is the most common mechanism for attention shifts, although all three occur. A typical thread of control can be followed from the initial receipt of a sensor datum. When a sensor receives contact, the fact is (eventually, after all sensor processing) POSTed on the target blackboard by the sensor contact perceptual demon as a "New Contact" message. Once this is posted, the TACCO will instantiate the trigger for the Identify Area of Interest task:

> "any new [sensor type] contact posted on contact level of target BB"

If no higher priority task (such as placing an attack) is active, this trigger will fire and allow Identify Area of Interest to capture control and execute. As the TACCO performs this task, that individual will POST several inferences to the target blackboard, the first being a possible DP AOI. Depending on the rest of the blackboard contents, this may instantiate the trigger for the Broaden Initial Contact task:

> "new DP AOI POSTed on AOI level of target BB AND NOT prudential pattern around contact buoy posted on pattern level of situation blackboard AND NOT strong directional hypothesis on directional level of target BB"

This complex trigger contains some implicit priority structure. When there is a strong directional hypothesis, the prosecution is already in an advanced state. In such cases, the TACCO is trying to refine the location, not just to broaden the contact, so a prudential pattern is not indicated. Similarly, if the pattern is already planned or deployed, there is no need to do it again. Thus, when the prosecution is in an early state and the task has not already been started or completed, the POSTing of the DP AOI by the Identify Area of Interest task will allow this task to capture attention from Identify Area of Interest.

Once it has attention, the Broaden Initial Contact will begin by subrogating to the Position AC in AOI task. This reflects that fact that the aircraft should be directed to cease its current movements and start the trip to the AOI before the actual plotting of the prudential pattern begins, as the lag time is potentially large. Once the Position AC in AOI task completes executing, and assuming no other sensor event has occurred that would cause attention to be captured by some other task (such as Identify AOI), the Broaden Initial Contact task would regain attention and resume at the point where it had subrogated to Position AC to AOI. At a point later in the task, attention might again be subrogated to another task, Plot Acoustic Environmental Features. This

would be done only if the dimensions of the DP area were not already plotted, which is actually the likely case in this example. If the subrogation occurs and control eventually returns to the Broaden Initial Contact task, then the actual FTPs for the prudential pattern would be laid in and pattern posted on the situation board.

Still assuming no further sensor events had changed the blackboard patterns by the end of the Broaden Initial Contact task, attention would be open to contention at the end of this task. The Identify AOI task, which had been interrupted by Broaden Initial Contact, would likely now regain attention and continue executing at the point where it was interrupted. Its next operations would be to post CZ AOIs on the target blackboard, beginning with the 2nd CZ and then the 1st CZ, in the case of a 2CZ environment. Either of these POSTings would create a new blackboard pattern that would instantiate the trigger for the Investigate Convergence Zones task:

CZ1 AOI or CZ2 AOI on target BB AND no DP hypothesis on target BB.

Here again, this trigger contains some implicit priority structure. Once the TACCO is working from a specific DP or locational hypothesis about the target, convergence zone contact will become of little interest and that individual will instead focus on developing and refining a series of direct path contacts. Thus, in this example, attention would be captured by this task and a CZ Investigative pattern would be planned.

Control would continue to flow between various tasks in this manner. After the CZ investigative pattern were planned and mapped out as FTPs, control would again be open to competition. Assuming the aircraft had not yet arrived at the location to deploy the prudential pattern, the TACCO would have time free to resort to various background tasks, such as Review Overall Situation or Manage Sonobuoy Resources. Eventually, the aircraft will approach the first fly-to-point for the prudential pattern, and the trigger for the DEPLOY SENSOR/PATTERN task will be instantiated:

"new planned pattern posted on situation BB AND expected arrival at next expendable FTP < threshold time"

In other words, as the aircraft approaches within some threshold of the first FTP for the pattern, the TACCO will respond to the need to arm and cue the sonobuoys to be deployed in the pattern.

# 5. SUMMARY AND CONCLUSIONS

This report has described the development of a domain specific model of human supervisory control in a real-world real-time multi-tasking domain, that of Naval Air ASW. It has also described the generalization of this model to a domain-independent framework and formal description language for building similar models in other domains. Thus, these research results have significance both by providing a new human-computer interaction modeling methodology and by constructing and validating a model of TACCO expertise which can be used in the development of future Air ASW systems.

This modeling framework, which is called COGNET, appears to have broad applicability to modeling human-computer interaction in real-time multi-tasking domains, an area in which existing modeling techniques were insufficient. Methodologically, COGNET represents a significant integration of two major techniques for modeling human problem solving, the primarily procedural and goal-directed control approach of GOMS and the primarily declarative and opportunistic control approach of blackboard systems. This integrated framework allows the capturing of goal-directed, data-directed, and opportunistic control, as well as the effect of the user's mental model (formalized as a blackboard representation) on his/her procedures for accomplishing goals in RTMT task domains.

COGNET provides a basis for much additional research regarding attention allocation, individual differences in problem-solving, and expertise development, particularly as they relate to human-computer interface and decision aid development As suggested in Zachary (1989), COGNET provides a new methodology for analyzing attention switching. Importantly, this method allows the effect of problem context -- as indicated by the pattern of information on the problem blackboard --on attention to be identified, modeled, and quantified.

In addition, the COGNET framework generates complete, computable representations of human problem solving in RTMT domains. Another major potential application of this framework is to build a specific COGNET model into a human-computer interface to give the computer a model of the system user. With this model, the resulting interface could observe the actions of the user and match them against the task models to identify and reason about the actions the user was tasking. By simulating the various perceptual demons and the effect of both the demons and the task models on the blackboard problem representation, the interface could develop a highly accurate model of what the user is doing, what the user might need to do next, and even estimate what interactive tasks could be productively automated for the user at any given point in the interaction. The development of this type of adaptive intelligent interface is currently underway as the next phase of this research. In the case of the Air ASW/TACCO model, this resulting adaptive interface could lead to improved TACCO-station interfaces for existing ASW platforms, specifically the P-3C baseline and updates I, II and III.

The COGNET models of ASW also have application value beyond the adaptive interface concept. A strength of the GOMS model (and its extensions in COGNET) is the clear separation of goals and operators/methods. The operators, particularly the behavioral ones (e.g. keystroking), are susceptible to change as systems, software and hardware evolve. The goal structure, however, captures the problem solving

approach of the person, independently from the specific means used to implement the approach (i.e; separate from the specific computer-human interface). A user who is a task expert might not change his/her goal hierarchy much when performing the task with different interfaces, even though her/his set of operators might change drastically. This is because much of her/his representation of the task is 'compiled' in a goal structure. In fact, Card et al's (1983) study of text editing demonstrates precisely that point -- they found the same goal structure applied across many different text editor interfaces. A COGNET task model therefore incorporates features of both the procedure and the representation aspects of human information processing:

- at the procedural level, a GOMS model includes the general problem-solving procedure via the goal hierarchy, and an interface-specific task performance procedure via the lowest level goals and attendant operators:
- at the representation level, a GOMS model incorporates compiled aspects of the underlying representation of the problem and task via the specific goals and goal decomposition it incorporates.

We view the COGNET task description framework as a highly general way of capturing and separating the general and interface-specific aspects of knowledge that are involved in expert-level human-computer interaction.

This view translates into some pragmatic benefits of the COGNET application to Air ASW Mission Management. The individual task models that make up the COGNET mission management model show a well-organized goal/subgoal structure. This structure captures much of the problem-solving strategy of the TACCO in the portions of the mission that we have characterized as "Mission management". Because this structure is independent of the individual workstation interface and is also highly dependent on the evolved expertise of the TACCOs, we feel that the general goal structure of the individual task models is likely to remain invariant even over major reconfigurations of the workstations or software, such as is intended for the Update IV version of the P-3C. Having a generic model of TACCO expertise should be of substantial significance in the development of:

- user software interfaces,
- crewstation layout and functionality, and
- training materials and manuals

for present and future Air ASW systems.

## APPENDIX

**GOAL: POSITION IN AREA OF INTEREST...*(new AOI posted on AOI level of target BB) AND NO (target hypothesis on DP or Location levels of target BB) OR (rules about what pattern currently monitoring, etc. later in mission)***

GOAL: Enter FTP for new AOI
Hook point in AOI
Determine type of FTP
Perform DES FTP (type, hook point)...*if*
Perform UNDES FTP (type, hook point)...*if*
Post "FTP [type, location]" on tactical display level of situatuion BB
Post "arrive in AOI about time [mission time]"on expectations level of Situation BB

GOAL: Clear unneeded FTP...*repeat until all unneeded FTPs cleared*
Hook FTP
Perform CLEAR DATA (FTP)
Unpost FTP from tactical display level of situation BB

GOAL: Adjust FTP pattern for time lag...*if (bearing shift posted on contact level of target BB) OR ((target transiting hypothesis on mission factors level of situation BB) AND (time since FTP entry > threshold time))*
use **FTP Adjustment Method...** *based on individual variation and amount of time before FTP capture*
**Method A: New Desired FTP**
GOAL: Enter new FTPs...*repeat for each FTP of pattern*
Hook adjusted FPT location
Perform UNDES FTP (type, location)...*if old FTP was undesignated*
Perform DES FTP...*if old FTP was designated*
Post "FTP [type, location]" on tactical display level of situatuion BB
GOAL: Clear old FTPs...*repeat for each FTP of pattern*
Hook FTP
Perform CLEAR DATA (FTP)
Unpost FTP from tactical display level of situation BB
**Method B: Higher Priority FTP**
GOAL: Enter new FTP
Hook adjusted FPT location
Perform UNDES FTP (expendable, location)...*if old FTP was undesignated*
Perform DES FTP (weapon, location)...*if old FTP was designated*
Post "FTP [type, location]" on tactical display level of situatuion BB
GOAL: Clear old FTPs...*repeat for each FTP of pattern*
Hook FTP
Perform CLEAR DATA (FTP)
Unpost FTP from tactical display level of situation BB

**GOAL: PLOT ACOUSTIC ENVIRONMENTAL FEATURES...*(new AOI posted on target BB) AND NOT (AOI already plotted)***

GOAL: Plot DP area..*if (DP AOI posted on AOI level of target BB) AND NOT (DP AOI already plotted)*

Hook contact buoy
Determine DP radius from mission factors level of situation BB
Perform DRAW CIRCLE (DP radius)
Post "DP circle around buoy [buoy number]" on tactical display level of situation BB

GOAL: Plot convergence zone...*if (CZ AOI posted on AOI level of target BB) AND (no DP hypothesis posted on DP level of target BB)*

Determine CZ radius and width from mission factors level of situation BB

GOAL: Plot CZ radius

Hook contact buoy
Perform DRAW CIRCLE (CZ radius)
Post "CZ radius circle around buoy [buoy number]" on tactical display level of situation BB

GOAL: Plot CZ width

Use **Plotting Method**...*based on individual variation*
Methods:

**Method A: Boundary Circles**

Hook contact buoy
Perform DRAW CIRCLE (CZ radius - 1/2 CZ width)
Hook contact buoy
Perform DRAW CIRCLE (CZ radius + 1/2 CZ width)
Post "CZ boundary circles around buoy [buoy number] on tactical display level of situation BB

**Method B: Width Circle**

Hook CZ/LOI Intersection
Perform DRAW CIRCLE (CZ width)
Post "CZ width circle for CZ associated with buoy [buoy number]" on tactical display level of situation BB

**Method C: Visual Estimation**

Determine estimated CZ inner and outer boundaries, based on CZ width from mission factors level of situation BB and tacplot scale from tactical display level of situation BB

**GOAL: REVIEW OVERALL SITUATION...*(off-screen display on situation BB contains many buoys) AND (time since last review > ??)***

GOAL: View full sonobuoy field
Perform UPSCAL...*repeat until full field in view*
Post "situation review at 'time'" on Situation BB

GOAL: Refocus display on AOI
Hook center of AOI...*if not near center of display*
Perform RECENTER (around hook point)...*if needed*
Perform DOWNSCAL...*repeat until AOI fills visible screen*

**GOAL: MANAGE SONOBUOY RESOURCES...*(new planned pattern on situation BB) OR (buoy resources low) OR ??***

GOAL: Determine number of buoys of desired type remaining
<u>Determine</u> whether currently displayed ARO page has desired information
<u>Perform</u> MORE...*if desired type not on currently displayed ARO page*
<u>Determine</u> number of buoys from ARO

**GOAL: BROADEN INITIAL CONTACT...*(new DP AOI posted on AOI level of target BB) AND NOT (prudential pattern on contact buoy on pattern level of situation BB) AND NOT (directional hypothesis on direction level of target BB)***

GOAL: Get AC to contact buoy
Subrogate to "Position AC in AOI"

GOAL: Focus tactical display on AOI
Perform UPSCAL...*until AOI visible on display*
Perform RECENTER (contact buoy)...*if contact buoy not near center of screen*
Perform DOWNSCAL...*until AOI fills display*

GOAL: Build prudential pattern
Subrogate to "Plot Acoustic Environmental Features"...*if DP not already plotted*
GOAL: Enter FTPs
Hook 1st point on DP circle
Perform UNDES FTP (expendable, hooked location)
Hook 2nd point on DP circle
Perform UNDES FTP (expendable, hooked location)
Hook 3rd point on DP circle
Perform UNDES FTP (expendable, hooked location)
Hook 4th point on DP circle
Perform UNDES FTP (expendable, hooked location)

Post "FTPs at [locations] on tactical display level of situation BB

Post "prudential pattern around contact [buoy number] planned" on pattern level of situation BB

Determine expected time of arrival at 1st FTP of pattern (distance from current aircraft location to FTP/ aircraft speed)

Post "Expected arrival at next FTP at 'time'" on expectations level of Situation BB

**GOAL: INVESTIGATE CONVERGENCE ZONES...*(CZ1 or CZ2 AOI on target BB) AND (no DP hypothesis on target BB)***

GOAL: Locate CZ2 on display...*if 2CZ on situation BB AND CZ2 AOI posted on target BB*

Determine CZ line of interest (LOI) definition (if only one contact with bearing steady, then LOI = contact buoy bearing; if more than one contact, then LOI = average of contact bearings; if one contact and bearing shift, then LOI= average of bearings)

Subrogate to"Plot Acoustic Environmental Features"...*if CZ not already plotted*

GOAL: Mark LOI/CZ intersection...*optional, more likely if LOI definition is not = contact bearing or if LOI does not intersect CZ midline*

*use **Marking Methods**...based on individual variation*

Methods:

**Mark LOI**

Hook contact buoy or average point of bearing lines near contact buoys

Perform DRAW LINE

Hook end of bearing line or average point of ends of contact bearings

**Mark POI**

Hook intersection of LOI and CZ midline

Perform REFMARK

GOAL: Focus tactical display on CZ2 AOI...*if (AOI not visible on screen) or (scale >128) or (scale <4)*

use **ScreenScaling Method**... *based on individual variations*

Methods:

**Method A: Center between contact buoy and CZ2/LOI intersection**

**Method B: Center on contact buoy with CZ2/LOI in view**

**Method C: Center on CZ2/LOI with contact buoy in view**

**Method D: Zoom in on CZ2/LOI**

GOAL: Build 2nd CZ investigative pattern

Post "2nd CZI Pattern around buoy 'contact buoy number' planned" on patterns level of Situation BB

use **Layout CZ2 Pattern Method**... *based on individual variation and mission factors: buoy resources (BR), confidence in existence of CZ in question (C), time remaining on station (TOS)*

Selection Rules:

If C high, BR high, and TOS either high or low; then use **Method A** (probability .5) or **Method B** (probability .5)

If C high and BR and TOS low; then use **Method C**

If C low, TOS high and BR high; then use **Method C**

If C low, TOS high, but BR low; then use **Method D**

Methods:

**Method A: Standard Line**

**Method B: Standard Line with Wings**

**Method C: Reduced 2-buoy**
**Method D: Reduced 1-buoy**

Transform "2nd CZI pattern around Buoy 'contact buoy number' planned" to "2nd CZI Pattern around buoy 'contact buoy number' mapped" on patterns level of situation BB

Post "FTPs [type, location]" on tactical display level of situation BB

GOAL: Locate CZ1 on display...*ifor all CZ1 AOIs on target BB for which there is no 1st CZI Pattern posted on situation BB*

Determine CZ line of interest (LOI) definition

Subrogate to"Plot Acoustic Environmental Features"...*if CZ not already plotted*

GOAL: Mark LOI/CZ intersection...*optional, more likely if LOI definition is not = contact bearing or if LOI does not intersect CZ midline*

*use **Marking Methods**...based on individual variation*

Methods:

**Mark LOI**

Hook contact buoy or average point of bearing lines near contact buoys

Perform DRAW LINE

Hook end of bearing line or average point of ends of contact bearings

**Mark POI**

Hook intersection of LOI and CZ midline

Perform REFMARK

GOAL: Focus tactical display on CZ1 AOI...*if (AOI not visible on screen) or (scale >128) or (scale <4)*

use **ScreenScaling Method**... *based on individual variations*

Methods:

**Method A: Center between contact buoy and CZ/LOI intersection**
**Method B: Center on contact buoy with CZ/LOI in view**
**Method C: Center on CZ/LOI with contact buoy in view**
**Method D: Zoom in on CZ/LOI**

GOAL: Build 1st CZ investigative pattern

Post "1st CZI Pattern around buoy 'contact buoy number' planned" on patterns level of Situation BB

use **Layout CZ Pattern Method**... *based on individual variation and mission factors: buoy resources (BR), confidence in existence of CZ in question (C), time remaining on station (TOS)*

Selection Rules:

If C high, BR high, and TOS either high or low; then use **Method A** (probability .5) or **Method B** (probability .5)

If C high and BR and TOS low; then use **Method C**

If C low, TOS high and BR high; then use **Method C**

If C low, TOS high, but BR low; then use **Method D**

Methods:

**Method A: Standard Line**
**Method B: Standard Line with Wings**
**Method C: Reduced 2-buoy**

**Method D: Reduced 1-buoy**

<u>Transform</u> "1st CZI pattern around Buoy 'contact buoy number' planned" to "1st CZI Pattern around buoy 'contact buoy number' mapped" on patterns level of situation BB

<u>Post</u> "FTPs [type, location]" on tactical display level of situation BB

**ScreenScaling Methods**

**Method A: Center between contact buoy and CZ2/LOI intersection**

<u>Perform</u> UPSCAL...*until both contact buoy and CZ/LOI intersection visible on screen*

<u>Perform</u> RECENTER (point approximately half way between contact buoy and CZ/LOI intersection)

<u>Perform</u> DOWNSCAL...*until area including contact buoy and CZ/LOI intersection fills screen*

**Method B: Center on contact buoy with CZ2/LOI in view**

<u>Perform</u> UPSCAL...*until both contact buoy and CZ/LOI intersection visible on screen*

<u>Perform</u> RECENTER (contact buoy)

<u>Perform</u> DOWNSCAL...*until area including contact buoy and CZ/LOI intersection fills screen*

**Method C: Center on CZ2/LOI with contact buoy in view**

<u>Perform</u> UPSCAL...*until both contact buoy and CZ/LOI intersection visible on screen*

<u>Perform</u> RECENTER (CZ/LOI intersection)

<u>Perform</u> DOWNSCAL...*until area including contact buoy and CZ/LOI intersection fills screen*

**Method D: Zoom in on CZ2/LOI**

<u>Perform</u> UPSCAL...*until both contact buoy and CZ/LOI intersection visible on screen*

<u>Perform</u> RECENTER (CZ/LOI intersection)

<u>Perform</u> DOWNSCAL...*until CZ AOI fills screen*

**Layout CZ Pattern Methods**

**Method A: Standard Line**

<u>Hook</u> point on LOI at CZ inner radius

<u>Perform</u> UNDES FTP (expendable, hooked location)

<u>Hook</u> point on LOI at middle of CZ

<u>Perform</u> UNDES FTP (expendable, hooked location)

<u>Hook</u> point on LOI at CZ outer radius

<u>Perform</u> UNDES FTP (expendable, hooked location)

**Method B: Standard Line with Wings**

<u>Hook</u> point on LOI at CZ inner radius

<u>Perform</u> UNDES FTP (expendable, hooked location)

<u>Hook</u> point on LOI at middle of CZ

<u>Perform</u> UNDES FTP (expendable, hooked location)

<u>Hook</u> point on LOI at CZ outer radius

<u>Perform</u> UNDES FTP (expendable, hooked location)

<u>Hook</u> point on CZ midline, 1 CZ width away from middle FTP, clockwise or counterclockwise

<u>Perform</u> UNDES FTP (expendable, hooked location)

Hook point on CZ midline, 1 CZ width way from middle FTP opposite direction of most recent FTP

Perform UNDES FTP (expendable, hooked location)

**Method C: Reduced 2-buoy**

Hook point on LOI at CZ inner radius

Perform UNDES FTP (expendable, hooked location)

Hook point on LOI at CZ outer radius

Perform UNDES FTP (expendable, hooked location)

**Method D: Reduced 1-buoy**

Hook point on LOI at middle of CZ

Perform UNDES FTP (expendable, hooked location)

**GOAL: DEPLOY SENSOR/PATTERN...*(new planned pattern posted on Situation BB) AND (expected arrival at next expendable FTP < 'threshold time')***

GOAL: Disarm unneeded buoys
Perform SONO DIS

GOAL: Arm sonobuoy pattern
Perform SONO SEL (type, life/depth)...*repeat for all planned buoys*

**GOAL: IDENTIFY AREA OF INTEREST...*any new [sensor type] contact posted on contact level of target BB***

GOAL: Identify Acoustic AOI...*if new DIFAR, LOFAR, CASS, or DICASS contact*

GOAL: Identify passive AOI...*if new DIFAR or LOFAR contact*

<u>Determine</u> [contact buoy number] from contact level of target BB

<u>Post</u> "DP AOI on 'contact buoy number' on AOI level of target BB

<u>Post</u> "BTmBn AOI on 'contact buoy number' on AOI level of target BB...*if Bottom Bounce on mission factors level of situation BB"*

<u>Post</u> "CZ1 AOI on 'contact buoy number' on target BB...*if 2CZ or 1CZ on mission factors level of situation BB*

<u>Post</u> "CZ2 AOI on 'contact buoy number' on target BB...*if 2CZ on mission factors level of situation BB*

<u>Transform</u> "New [buoy type] contact at time [mission-time] on Buoy [Channel No] with Bearing [bearing]" into "[buoy type] contact at time [mission-time] on Buoy [Channel No] with Bearing [bearing]"on contact level of target BB

GOAL: Identify active AOI...*if CASS or DICASS contact*

<u>Determine</u> [contact buoy number] from contact level of target BB

<u>Post</u> "CASS AOI with MDR [distance] on buoy [Channel No]" on AOI level of target BB...*if CASS contact*

<u>Post</u> "DICASS AOI with bearing [bearing] and with MDR [distance] on buoy [Channel No]" on AOI level of target BB...*if DICASS contact*

<u>Transform</u> "New [buoy type] contact at time [mission-time] on Buoy [Channel No] with Bearing [bearing]" into "[buoy type] contact at time [mission-time] on Buoy [Channel No] with Bearing [bearing]"on contact level of target BB

GOAL: Identify Non-acoustic AOI...*if MAD or RADAR contact*

GOAL: Identify RADAR AOI...*if RADAR contact*

<u>Post</u> "RADAR AOI with MDR [distance] at location [location]" on on AOI level of target BB

<u>Transform</u> "New RADAR contact at time [mission-time] at location [location]" into "RADAR contact at time [mission-time] at location [location]"on contact level of target BB

GOAL: Identify MAD AOI...*if MAD contact*

<u>Post</u> "MAD AOI with MDR [distance] at location [location]" on AOI level of target BB

<u>Transform</u> "New MAD contact at time [mission-time] at location [location]" into "MAD contact at time [mission-time] at location [location]"on contact level of target BB

GOAL: Abort pattern...*if true contact*

<u>Subrogate</u> to "Get AC to AOI"

## FUNCTION METHODS for all multistep functions

### GOAL: Perform SONO SEL

Perform SONO SEL

Select D1...*if DIFAR desired*

GOAL: Select life/depth for DIFAR buoy...*if DIFAR selected*
- Select D1...*if short/shallow desired*
- Select D2...*if shor/deep desired*
- Select D3...*if long/shallow desired*
- Select D4...*if long/deep desired*

Select D2...*if DICASS desired*

GOAL: Select depth...*if DICASS selected*
- Select D1...*if shallow desired*
- Select D2...*if deep desired*

### GOAL: Perform WEAP SEL

Perform WEAP SEL

GOAL: Select weapon depth
- Select D1...*if 250 feet desired*
- Select D2...*if 500 feet desired*
- Select D3...*if 750 feet desired*
- Select D4...*if 1000 feet desired*

### GOAL: Perform DES FTP

Hook FTP desired location

Perform DES FTP

Select D1...*if weapon FTP desired*

Select D2...*if expendable FTP desired*

Select D3...*if normal FTP desired*

Select D4...*if monitor FTP desired*

Enter bearing

**GOAL: Perform UNDES FTP**

<u>Hook</u> FTP desired location

<u>Perform</u> UNDES FTP

<u>Select</u> D1...*if expendable FTP desired*

<u>Select</u> D2...*if orbit FTP desired*

<u>Select</u> D3...*if normal FTP desired*

<u>Select</u> D4...*if monitor FTP desired*

<u>Enter</u> orbit radius...*if orbit FTP selected*

**GOAL: Perform DES FIX**

<u>Hook</u> point of hypothesized target location

<u>Perform</u> DES FIX

**GOAL: Perform AC CNTRL**

<u>Perform</u> AC CNTRL

<u>Enter</u> new altitude or speed

**GOAL: Perform GEN TRK**

<u>Perform</u> GEN TRK

<u>Hook</u> first (earliest) fix or MAD symbol

<u>Hook</u> second fix or MAD symbol

**GOAL: Perform CLR TRK**

<u>Perform</u> CLR TRK

<u>Enter</u> number of track symbol to be cleared

**GOAL: Perform DRAW CIRCLE**

<u>Hook</u> desired location of center of circle

Perform DRAW CIRCLE

Enter radius of circle...*if desire circle of a known radius*

Hook desired location of any point on circle circumfrence...*if radius not entered*

**GOAL: Perform DRAW LINE**

Hook point indicating one end of desired line

Perform DRAW LINE

Enter length and bearing of line...*if desire line of known length and bearing*

Hook point indicating other end of desired line...*if length and bearing not entered*

## REFERENCES


Anderson, J.R. & Bower, G.H. (l973). Human associative memory. Washington, DC: V.H. Winston and Sons.

Birmingham, H.P. & Taylor, J. (l954). A design philosophy for man-machine control systems. Proceedings IRE, 42, l748-1758.

Bovair,S., Kieras, D.E., & Polson, P.G. (1988). The acquisition and performance of text-editing skill: A production-system analysis (Technical Report No. 28). Ann Arbor: University of Michigan.

Buchanan, B. G., & Shortliffe, E. H. (Eds.). (1984) Rule-based expert systems: The MYCIN experiments of the Stanford heuristic programming project. Reading, Ma: Addison-Wesley.

Card, S., Moran, T., & Newell, A. (l983). The psychology of computer-human interaction. Hillsdale, NJ: Lawrence Erlbaum Press.

Chubb, G. P., Laughery Jr., K. R., & Pritsker, A. B. (1987) Simulating Manned Systems. In Handbook of Human Factors. New York, NY: Wiley & Sons.

Croft, B., Lefkowitz, L., Lesser, V., & Hufff, K. (1983). Interpretation and planning as a basis for intelligent interface. In Proceedings of the Conference on Artifical Intelligence. Rochester, MI: Oakland University.

Durfee, E.H., Lesser, V.R., & Corkill, D.D. (1989). Cooperation through communication in a distributed *problem solving network. In S. Robertson,* W. Zachary, & J. Black (Eds.), Cognition, computation and cooperation. Norwood, NJ: Ablex Press.

Elkerton, J., & Palmiter, S. (1989). Designing help systems using a GOMS model: Part 1 - An information retrieval evaluation (Technical Report C4E-ONR-3). Ann Arbor: University of Michigan, Center for Ergonomics.

Ericsson, K., & Simon, H. A., (1980). Verbal reports as data. Psychological Review, 87, 215-251.

Ericsson, K., & Simon, H. A., (1984). Protocol analysis: Verbal reports as data. Cambridge, MA: MIT Press.

Erman, L.D., Hayes-Roth, F., Lesser, V.R., & Reddy, D.R. (1980) The Hearsay-II speech understanding system: Integrating knowledge to resolve uncertainty. Computing Surveys, 12, 213-253.

Fitts, P.M. (l954) The information capacity of the human motor system in controlling amplitude of movement. Journal of Experimental Psychology, 47, 38l-39l.

Glenn, F. G. (1988) Development of a human operator simulator version V (HOS-V): Concept formulation. (Technical Report 880430.8703), Blue Bell, PA: CHI Systems Inc.

Goodson, J.L., & Schmidt, C.F. (1989) The design of cooperative person-machine problem solving systems: A methodology and our example. In S. Robertson, W. Zachary, & J. Black (Eds.), Cognition, computation and cooperation. Norwood, NJ: Ablex Press.

Hayes-Roth, B. (l983) A Blackboard Model of Control. Technical Report, Computer Science Department, Stanford University, Stanford, CA.

Hayes-Roth, B. & Hayes-Roth, F. (l979) "A Cognitive Model of Planning." Cognitive Science, 3, 275-3l0.

Hewitt,C.E., The Apiary Network Architecture for Knowledgeable Systems, Conference Record of the 1980 Lisp Conference, 1980, Stanford University, Stanford, CA

Hopson, J. & Zachary, W. (l982) "Experiences with computer aided aircrew decision making." S.A.E. Transactions, 9l, 4388-4402.

Johannsen, G. (1976) Preview of man-vehicle control session, in T. Sheridan and G. Johannsen, eds. Monitoring behavior and supervisory control. New York: Plenum.

John , B. & Newell, A. Predicting the Time to Recall Computer Command Abbreviations. Proceedings of CHI+GI 1987. 33-40. ACM: New York. 1986

John, B., Rosenbloom, P. & Newell, A. A Theory of Stimulus-Response Compatibility Applied to Human-Computer Interaction. Proceedings of CHI '85. 213-219. ACM: New York. 1985

Kieras, D.E. (1988). Towards a practical GOMS model methodology for user interface design. In M. Helander (Ed.), Handbook of human-computer interaction. Amsterdam: Elsevier.

Lesser, V.R., & Corkill, D.D., (1981) Functionally-accurate, cooperative distributed systems. IEEE Transactions on Systems, Man and Cybernetics, SMC-11(1): 81-91.

Malone, T.W., (1989). Organizing information processing systems: Parallels between human organizations and computer systems. In S. Robertson, W. Zachary, and J. Black (Eds.), Cognition, computation and cooperation. Norwood, NJ: Ablex Press.

Moran, T.P. (l98l) "The Command Language Grammar: A Representation for the User Interface of Interactive Computer Systems." International Journal of Man-Machine Studies, l5, 3-50.

Newell, Allen, Some problems of basic organization in problem-solving programs, M.C. Yovits, G.T. Jacobi, G.D. Goldstein, 1962, 393-423, Spartan Books, Washington D.C.

Newell, A., Shaw, J.C., & Simon, H. (l959) "Report on a General Problem-Solving Program." Proceedings of the International Conference on Information Processing (UNESCO), Paris.

Newell, A. & Simon, H.A. (1972) Human Problem Solving . Englewood Cliffs, N.J.: Prentice-Hall.

Nii, P. H., Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures (PART ONE), AI Magazine, 1986, 7, 2, 38-53,

Nii, P. H., Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures (PART TWO), AI Magazine, 1986, 7, 3, 82-106,

Norcio, A.F., & Stanley, J. (1989). Adaptive human-computer interfaces: A literature survey and perspective. IEEE Traqnsactions on Systems, Man, and Cybernetics, 19(2), 399-408.

Norman, D. (1986) Cognitive engineering. In D.A. Norman & S.W. Draper (Eds.), User centered system design. Hillsdale, NJ: Erlbaum.

Norman, D. (l983) Some observations on Mental models. In D. Gentner, and A.L. Stevens eds., Mental models, Hillsdale, NJ: Erlbaum.

Pylyshyn, Z. (l984) Computation and cognition. Cambridge, MA: The MIT Press.

Quillian, M.R. (l968) Semantic Memory, in M. Minsky Ed., Semantic Information Processing, Cambridge, MA: MIT Press.

Rasmussen, J. (1986). A framework for cognitive task analysis in systems design. In E. Hollnagel, G. Mancini, & D.D. Woods (Eds.), Intelligent decision support in process environments. Berlin: Springer-Verlag.

Rasmussen, J. (1986b). Information processing and human-machine interaction. New York: North-Holland.

Robertson, S., Zachary, W., & Black, J. (Eds.), (1989) Cognition, computation and cooperation. Norwood, NJ: Ablex Press.

Rumelhart, D. E., McClell&, J. L. and the PDP Research Group (1986). Parallel distributed processing: Explorations in the microstructure of cognition (Volume 1: Foundations). Cambridge, MA: MIT Press.

Sacerdoti, E.D. (l974) Planning In a Hierarchy of Abstraction Spaces. Artificial Intelligence, 5, ll5-l35.

Schwartz, J.P. & Jamar, P. (l983) Lack of guidance for decision aid interface design Association for Computing Machinery SIGCHI Bulletin, l5, l3-7.

Selfridge, Oliver G., Pandemonium: A paradigm for learning, Proceedings of the Symposium on the Mechanization of Thought Processes, 1959, 511-529,

Waterman, D. A. , A Guide to expert systems, Reading, Mass.: Addison-Wesley, 1986.

Wohl, J.G., Entin, E.E., & Eterno, J.S. (l983) Modeling human decision processes in command and control. (TR-l37), Burlington, MA : Alphatech, Inc.

Woods, D.D., & Hollnagel, E. (1987). Mapping cognitive demands in complex problem-solving worlds. International Journal of Man-Machine Studies, 26, 257-275.

Woods, D.D., & Roth, E.M. (1988). Cognitive systems engineering. In M. Helander (Ed.), Handbook of human-computer interaction. Amsterdam: Elsevier.

Zachary, W.W. (1989). A context-based model of attention switching in computer-human interaction domains. In Proceedings of the 33rd Annual Meeting of the Human Factors Society (pp. 286-290). Santa Monica, CA: Human Factors Society.

Zachary, W.W., & Zubritzky, M.C. (1988). An Experimental environment and laboratory for studying human information processing in Naval Air ASW (Technical Report 881020.8704). Blue Bell, PA: CHI Systems,Inc.

Zachary, W.W., & Zubritzky, M.C. (1989). A cognitive model of real-time data fusion in Naval Air ASW. In Proceedings of the 3rd Annual Data Fusion Symposium, Applied Physics Laboratory, JHU Laurel, MD.

Zachary,W., Zubritzky, M.,& Glenn F.(1988) "The Development of air antisubmarine warfare mission testbed as a tool for the development of operator models Proceedings of Human Factors Society 32nd Annual Meeting, Anaheim CA.

Zubritzky, M.C., Zachary, W.W., & Ryder, J.M. (1989). Constructing and applying cognitive models to mission management problems in air anti-submarine warfare. In Proceedings of the 33rd Annual Meeting of the Human Factors Society (pp. 129-133). Santa Monica, CA: Human Factors Society.

OUSDR (A)/R&A(E&LS)
Pentagon 3D129
Washington, DC 20301-3080

Dr. Bruce Barnes
Program Director, Div. Info
Robotics, & Intell. Systems
National Science Foundation
1800 G Street, N.W.
Washington, DC 20550

Mr. Charles Bates
Director, HE Division
USAF AAMRL/HE
WPAFB, OH 45433

Dr. Kenneth R. Boff
AAMRL/HE
WPAFB, OH 45433

Mr. Luiz Cabral
Naval Underwater Sys. Ctr.
Code 2212
Bldg. 1171/1
Newport, RI 02841

Cdr. Robert C. Carter USN
Naval Research Laboratory
Code 5532
Washington, DC 20375-5000

Dr. Marvin S. Cohen
Dec. Sci. Consortium, Inc.
1895 Preston White Drive
Suite 300
Reston, VA 22091

Dr. Stanley C. Collyer
Office of Naval Technology
Code 222
800 N. Quincy Street
Arlington, VA 22217-5000

Defense Tech. Info. Center
Cameron Station, Bldg 5
Alexandria, VA 22314
(2 copies)

Prof. Jay Elkerton
University of Michigan
Dept. of I & O Engineerg
Center for Ergonomics
1205 Beal Avenue
Ann Arbor, MI 48109-2117

Dr. Pamela Ebert Flattau
NAS-National Research Council
Committee on Vision
2101 Constitution Avenue, N. W.
Washington, DC 20418

Dr. Robert A. Fleming
Naval Ocean Systems Center
Human Factors Support Grp.
1411 South Fern Street
Arlington, VA 22202-2896

Dr. Alfred R. Fregly
AF Office of Sci. Res.
Life Sciences Directorate
Bldg. 410
Bolling AFB, DC 20332-6448

Jeffrey Grossman
Naval Ocean Systems Center
Code 4402
Bldg. 334
San Diego, CA 92152

Dr. Edgar M. Johnson
Technical Director
US Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. William B. Johnson
Search Technology, Inc.
4725 Peachtree Corners Circle
Suite 200
Norcross, GA 30092

Capt T. E. Jones, MSC, USN
Aviation Medicine & Human
Performance (Code 404)
Naval Medical R & D Cmd
National Capital Region
Bethesda, MD 21814-5044

Dr. Michael Kaplan,
Director, Office Basic Res
US Army Research Institute
5001 Eisenhower Avenue
Alexandria, VA 22333-5600

Dr. Steven W. Keele
University of Oregon
Department of Psychology
Eugene, OR 97403-1219

Dr. Arthur F. Kramer
University of Illinois
Department of Psychology
Champaign, IL 61820

Mr. Keith Kramer
Naval Research Laboratory
Code 5532
Washington, DC 20375-5000

Dr. David Le Berge
U. of California - Irvine
Cognitive Sciences Dept.
Irvine, CA 92717

Dr. Richard Loutitt
National Science Foundation
Division of Behavioral &
Neural Sciences
1800 G Street, N. W.
Washington, DC ^^550

Dr. Robert A. Lufti
Waisman Center
University of Wisconsin
Madison, WI 53706

Capt. William Moroney, USN
Naval Air Development Ctr.
Code 602
Warminster, PA 18974

Dean of the Academic Dept.
U. S. Naval Academy
Annapolis, MD 21402-5018

Naval Aerospace Medical
Research Laboratory
Sensory Division Code 23
Pensacola, FL 32508

Commanding Officer
Naval Air Systems Command
Crew Station Design
NAVAIR 5313
Washington, DC 20361

Commanding Officer
Naval Biodynamics Lab
Michoud Station
Box 29407
New Orleans, LA 70189

Commanding Officer
Navy Health Research Ctr.
P.O. Box 85122
San Diego, CA 92138

Commanding Officer
Navy Personnel R&D Center
San Diego, CA 92152-6800

Director
Technical Information Div.
Code 2627
Naval Research Laboratory
Washington, DC 20375-5000

US Naval Test Center
Aircrew Systems Branch
Systems Engineering Test
Directorate
Patuxent River, MD 20670

Commanding Officer
Naval Weapons Center
Human Factors Branch
Code 3152
China Lake, CA 93555

Dr. Michael Letsky
Office of the Chief of
Naval Operations
OP-01B7
Washington, DC 20350

Office of the Chief of
Naval Operations
OP-933D3
Washington, DC 20350-2000

Office of Naval Research
Special Asst. for Marine
Corps Matters
Code 00MC
800 N. Quincy Street
Arlington, VA 22217-5000

Office of Naval Research
Perceptual Science Program
Code 1142PS
800 N. Quincy Street
Arlington, VA 22217-5000
(3 copies)

Dr. Jesse Orlansky
Inst. for Defense Analyses
1801 N. Beauregard Street
Alexandria, VA 22311

Dr. W. A. Rizzo
Head, Human Factors Div.
Naval Training Systems Ctr
12350 Research Parkway
Orlando, FL 32826-3224

Dr. H. Wallace Sinaiko
Manpower Research & Advisory Services
Smithsonian Institution
801 N. Pitt St. Suite 120
Alexandria, VA 22314-1713

LCdr. Timothy Singer, USN
Naval Air Development Ctr.
Human Factors Engrg. Div.
Warminster, PA 18974

Mr. James G. Smith
Office of Naval Research
Code 121
800 N. Quincy Street
Arlington, VA 22217-5000

Cdr. S. Snyder
Office of Naval Technology
Code 222
800 N. Quincy Street
Arlington, VA 22217-5000

Mr. Nelson F. Steele
Advanced Res Dev Corp
9151 Rumsey Road
Columbia, MD 21045

Dr. John Tangney
AF Office of Sci. Res.
Life Sciences Directorate
Bldg. 410
Bolling AFB, DC 20332-6448

Dr. James T. Todd
Brandeis University
Psychology Department
Waltham, MA 02254

Dr. Douglas Towne
U. of Southern California
Behavioral Technology Lab
1845 South Elena Avenue
Fourth Floor
Redondo Beach, CA 90277

Dr. Harold P. Van Cott
Committee on Human Factors
NAS-National Res. Council
2101 Constitution Ave, N.W.
Washington, DC 20418

Dr. John Weisz
Technical Director
US Army Human Engineering
Laboratory
Aberdeen P. G., MD 21005

Prof. Christopher D. Wickens
Department of Psychology
University of Illinois
Champaign, IL 61801

SECURITY CLASSIFICATION OF THIS PAGE

# REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

| | |
|---|---|
| 1a REPORT SECURITY CLASSIFICATION: Unclassified | 1b RESTRICTIVE MARKINGS: N. A. |
| 2a SECURITY CLASSIFICATION AUTHORITY: N. A. | 3 DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution unlimited. |
| 2b DECLASSIFICATION/DOWNGRADING SCHEDULE: N. A. | |
| 4 PERFORMING ORGANIZATION REPORT NUMBER(S): Tech. Rep. 891215.8704 | 5 MONITORING ORGANIZATION REPORT NUMBER(S): Same |
| 6a NAME OF PERFORMING ORGANIZATION: CHI Systems, Inc. / 6b OFFICE SYMBOL (If applicable): | 7a NAME OF MONITORING ORGANIZATION: Office of Naval Research |
| 6c ADDRESS (City, State, and ZIP Code): Gwynedd Plaza III, Bethelem Pike and Sheble Lane, Spring House, PA 19477 | 7b ADDRESS (City, State, and ZIP Code): 800 N. Quincy Street, Arlington, VA 22217-5000 |
| 8a NAME OF FUNDING/SPONSORING ORGANIZATION: Office of Naval Research / 8b OFFICE SYMBOL (If applicable): Code 1142PS | 9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: N00014-87-C-0814 |
| 8c ADDRESS (City, State, and ZIP Code): 800 N. Quincy Street, Arlington, VA 22217-5000 | 10 SOURCE OF FUNDING NUMBERS |

| PROGRAM ELEMENT NO | PROJECT NO | TASK NO | WORK UNIT ACCESSION NO |
|---|---|---|---|
| 61153N 42 | RR04209 | 0420901 | R&T4429007 |

11 TITLE (Include Security Classification)
(U) A cognitive model of human-computer interaction in naval air ASW mission management

12 PERSONAL AUTHOR(S)
Zachary, W., Ryder, J. A., & Zubritzky, M. C.

| 13a TYPE OF REPORT | 13b TIME COVERED | 14 DATE OF REPORT (Year, Month, Day) | 15 PAGE COUNT |
|---|---|---|---|
| Technical | FROM 87/09/30 TO 90/03/30 | 15 December 1989 | 90 |

16 SUPPLEMENTARY NOTATION

| 17 COSATI CODES | | | 18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number) |
|---|---|---|---|
| FIELD | GROUP | SUB-GROUP | human-computer interaction; decision-making; cognitive model; ASW; COGNET; attention; GOMS |
| | | | |
| | | | |

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

Few, if any, techniques exist for cognitive modeling of human-computer interaction (HCI) in real-time multi-tasking (RTMT) problem domains, despite the fact that this class of problems includes many of the most critical and challenging person-machine systems. A new modeling formalism for representing HCI in RTMT domains is developed and applied to a realistic domain from Naval aviation, Airborne Antisubmarine Warfare (Air ASW) mission management. The formalism is called the Cognitive Network of Tasks or COGNET model with the blackboard architecture. The COGNET technique is used to model HCI in the example domain (Air ASW), resulting in a detailed and computable model of the strategies used by the human decision-makers. The model is built from data collected from expert human operators in a realistic experimental environment, described in a previous report.

| | |
|---|---|
| 20 DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS | 21 ABSTRACT SECURITY CLASSIFICATION: Unclassified |
| 22a NAME OF RESPONSIBLE INDIVIDUAL: John J. O'Hare | 22b TELEPHONE (Include Area Code): (202) 696-4502 / 22c OFFICE SYMBOL: Code 1142PS |

DD Form 1473, JUN 86 — Previous editions are obsolete — SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603